2007-04-01

From Amorphous to Defined: Balancing the Risks of Spiral Development

John Dillard

NPS-AM-07-002

ACQUISITION RESEARCH SPONSORED REPORT SERIES

From Amorphous to Defined: Balancing the Risks of Spiral Development

30 April 2007

by

John T. Dillard, Senior Lecturer
Graduate School of Business & Public Policy

David N. Ford, PhD, PE
Zachry Department of Civil Engineering
Texas A&M University

Approved for public release, distribution unlimited.

Prepared for: Naval Postgraduate School, Monterey, California 93943

The research presented in this report was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
e-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Abstract

The DoD's evolutionary acquisition policy is directed against project risk, but bears inherent risks of its own. The DoD policy for evolutionary acquisition mandates multiple product releases via spiral (i.e., amorphous & unplanned) or incremental (i.e., defined & deferred) development methodologies for all programs. All amorphous spirals eventually become definitive increments. Incremental development entails the deliberate deferral of work to a subsequent phase. Computational organizational modeling using systems dynamics reveals that this methodology introduces more concurrency during development, and more variety in production. The result is earlier delivery of the first increment, but with later and more costly delivery of subsequent increments than if conducted via a single-step methodology. Curtailments of scope by the exclusive use of mature technology enable more effective delivery of the first increment, further illustrated by two case studies. Duplication, rework, transaction costs, decision backlog and error are causes of inefficiency in the successive increments. Production variety and mixed configurations produce obvious implications for logistical supportability, training, failure causality, compatibility and interoperability, etc. Further, certain attributes of hardware products might help determine the suitability of this development methodology. Products that are nearly immutable, which have binary requirements for key capabilities, require man-rating, or are maintenance-intensive may not be good candidates for incremental development. Mutable products with costless production, continuous requirements, low maintenance, or time criticality are more likely to reap advantages from this development approach. While modular open systems architecture facilitates system adaptation, modularity itself does not necessarily create evolutionary advantages, due to relative modular interdependency. Program managers must be aware of the inherent risks of these agile acquisition methods and take additional steps to balance them with appropriate planning and resources, disciplined change-control measures, organizational accommodations and accountability for configuration management.

Keywords: Evolutionary acquisition, spiral development, incremental product development, Javelin, ATACMS, agile development methodologies, computational organizational modeling, modularity.

Acknowledgements

The authors wish to acknowledge the leaders of the NPS Acquisition Research Program: James B. Greene, RADM, USN, (Ret) and Dr. Keith Snider, as well as the tireless efforts of Karey Shaffer, her assistant David Wood and editorial team, without whose support we could not have conducted this research into the computational modeling of spiral development for Defense acquisition.

THIS PAGE INTENTIONALLY LEFT BLANK

About the Authors

John T. Dillard joined the NPS faculty in the fall of 2000 with extensive experience in the field of systems acquisition management. His research focuses on defense acquisition policy changes and their implications. Dillard began his career in program and contract management after attaining a MS in Systems Management from the University of Southern California in 1985. He has been involved with myriad technologies and system concepts that have evolved into fielded products: such as the M-4 Carbine, 120mm Mortar, and M-24 Sniper Weapon. He was the Assistant Project Manager for Research & Development of both the Army Tactical Missile System and the JAVELIN Anti-tank Weapon System at Redstone Arsenal, Alabama. He was the Product Manager for the Joint Advanced Special Operations Radio System, and in 1998 was appointed to head Defense Department contract administration in the New York metropolitan area. As an adjunct professor for the University of California at Santa Cruz, he teaches courses in project management and leadership to Silicon Valley public- and private-industry professionals.

John T. Dillard
Senior Lecturer, Naval Postgraduate School
Monterey, CA 93943-5197
Phone: (831) 656-2650
E-mail: jdillard@nps.edu

David N. Ford received his BS and MS from Tulane University, New Orleans, LA, and his PhD from the Massachusetts Institute of Technology, Cambridge. He is a Professor in the Construction Engineering and Management Program, Zachry Department of Civil Engineering, Texas A&M University, College Station. Prior this position, he was on the Faculty of the Department of Information Science, University of Bergen, Bergen, Norway, where he researched and taught in the System Dynamics Program. For over 14 years, he designed and managed the development of constructed facilities in industry and government. His current research interests include the dynamics of development projects and supply chains, strategic managerial flexibility, and resource-allocation policies. Dr. Ford is a member of INFORMS, ASCE, and other professional organizations.

David N. Ford, Ph.D., P.E.
Zachry Department of Civil Engineering
Texas A&M University
College Station, TX 77843-3136
Voice: (979) 845-3759
Fax: (979) 845-6554
Email: davidford@tamu.edu

THIS PAGE INTENTIONALLY LEFT BLANK

NPS-AM-07-002

ACQUISITION RESEARCH
SPONSORED REPORT SERIES

From Amorphous to Defined: Balancing the Risks of Spiral Development

30 April 2007

by

John T. Dillard, Senior Lecturer
Graduate School of Business & Public Policy

David N. Ford, PhD, PE
Zachry Department of Civil Engineering
Texas A&M University

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

Executive Summary

Our purpose in this research was to discover what spiral development really is, observe it in past programs, model it, and make predictions and recommendations for program managers. Program managers typically seek stability, in requirements, funding, system design, and production configuration. But it seems the only constant is change. Like the aspects of being temporary and unique, *progressive elaboration* is a project characteristic, and also a technique for incremental discovery of requirements and product utility.

Background

There are many new DoD terms for project management and product development methods. DoD promulgated *evolutionary acquisition* (EA) as policy in 2000, and soon after, *spiral development* for the preferred acquisition strategy of all materiel. EA's goal is to phase requirements and provide capability sooner. But there has been confusion over terms, despite further elaboration and even codification in statute, and it still persists today, along with a lack of full understanding of many policy implications – especially some inherent risks. EA operationally means there will always be multiple product releases of an item.

The policy thrust is primarily about the reduction of product cycle time within an uncertain environment, by exclusively using mature technology. DoD's requirements process has also followed with "evolutionary" requirements documents – a new idea. Uncertainty is the usual realm of program managers, especially in defense systems, and is usually dealt with by seeking best information. Earlier reform initiatives were aimed at overcoming information gaps and technology lag. For example, the 1990's Integrated Product and Process Development (IPPD) initiative was about gaining collective wisdom for early and complete requirements realization. However, the current paradigm is to allow uncertainty in requirements to resolve over time and endeavor only what is immediately achievable. The GAO has also urged the DoD to move toward Knowledge-Based Acquisition, with Technology Readiness Levels (TRL) as the rubric for program initiation (advanced development). Thus, the heart of EA is the exclusive use of mature technology to reduce project scope.

Questions about Policy Implications

EA outcomes are as yet unknown, and some authors have already had insightful strategic and institutional concerns. We have also had tactical (implementation) concerns about excessive decision bureaucracy, organizational challenges from multiple and concurrent development efforts, old technology at release, funds forecasting, transaction costs, and maintenance of subsequent increment priority.

Spiral development as a one-size-fits all strategy may not be appropriate. Variety adds complexity in production and is costly, for hardware owners and manufacturers alike. Both concurrency and variety are elements of program complexity and risk. Traditional views about late design changes are negative, except for producibility enhancements and savings. But market consumers often need items in rapid cycle times and appreciate product differentiation. In support of EA policy, the GAO has used product examples such as commercial vehicles, which ignore the aspect of ownership.

Control measures are used to manage risk. One way of coping with the complexities of variety in ownership is via organizational and individual accountability, and we use examples of these with illustrations of recent small arms variety and Rickover's nuclear Navy. Many other useful theorems on systems complexity, change and control exist.

More questions about spiral development include whether certain product characteristics determine spiral development method applicability. Mutability simplifies change, and spiral development was conceived for the most malleable of products: "soft" ware, which is virtually costless in production. This approach was to allay software project risk. And that idea can be extended in the case of DoD projects. Time criticality and life-saving dependency, as opposed to user hazard levels (safety & man-rating), might influence design approaches. We believe this is why space experts say they'll not use spiral development with NASA's new Crew Exploration Vehicle project. Regarding product size or production quantity, we find no evidence that either precludes use of spiral development – as with space vehicles and large ships -- though support considerations do arise with variety that could greatly affect total costs of ownership. Regarding "range of requirement attainment," binary key performance parameters could fall upon the critical path, making division into capability increments less beneficial. Increment phasing (the amount of concurrency) and cycle time (lead time) affect program complexity, budgeting, organizational stress, etc. Simon's views on complexity and evolution of systems involved hierarchy and modularity within architecture, but fail to emphasize modular interdependency. We cannot yet model these product attributes, but can illustrate most of them with examples from our case studies.

Development Case Studies

One of the most recent monographs we have found on emerging results of evolutionary acquisition is by RAND – on five immature, non-man-rated space systems. Space systems are somewhat different (in quantities, space environment, front-end investment, and extended technology development periods). RAND also found that policy confusion persists, and that EA added program complexity and uncertainty, especially with regard to budgeting. Extending their findings to non-space DoD programs, RAND highlighted the EA challenges of programmatic flux. They feel, and we agree, that EA presents the opportunity for typical PM challenges to be even more formidable.

Two missile programs were used as case studies for analysis and to illustrate planned and unplanned change. The Army Tactical Missile System (ATACMS) used both incremental and spiral strategies for product development. The program skipped its technology development phase by employing mature technologies for a leap-ahead capability in range. It arrived on budget and schedule, with several successive variants, pre-planned and unplanned. One instance of production change caused missile failure and costly refit of already produced missiles – underscoring the need for more thorough design specification and configuration management accountability.

Javelin used the single-step-to-full-capability approach to product development. The program embarked upon advanced development with immature technologies in several critical areas, causing significant cost and schedule overruns. It also has experienced subsequent design changes and product variety, more so as running production changes than as product variants.

Synthesis of these cases conveys that as an approach oriented primarily for reduction of product cycle time, spiral development can successfully be used when developing mature technologies first. But that a system's physical properties like mutability, along with other factors such as time criticality (user risk), and modular interdependency will drive spiral development applicability. And key capabilities may in fact depend upon the least mature technologies or even binary requirements, which we describe as attained/unattained (versus continuous). An "open," or at least elegant, architecture is key to form a basis for modular variety, and thorough design specification and configuration management accountability is essential for managing the complexity of multiple product releases. All amorphous spirals will eventually become defined increments, and even then may be popularly termed as "spirals." Other well-known programs have used a spiral approach over their long product life spans, but often having rather successive (versus highly concurrent) phasing of their development increments.

Computational Modeling of Spiral Development

Using system dynamics, our computational modeling of spiral versus a single-step methodology yields results that illustrate our implementation concerns. Spiral development can provide the initial increment delivery with some (but not all) requirements satisfied earlier than single-block development. However, spiral development takes more time and costs more to satisfy all requirements than single-block development. Spiral development has a high risk of not satisfying all requirements by the time single-block development can satisfy all requirements.

The concurrent use of multiple development blocks in spiral development significantly increases the number of development phases and activities that must be managed and coordinated at any given time compared to single-block development. This increases the project management needs for successful acquisition in spiral development projects when compared to single-block projects.

As in single-block development, progress in spiral development requires the identification and understanding of progress bottlenecks. The concurrence and resulting complexity of development in spiral projects causes the types and locations of bottlenecks to vary widely and be more difficult to identify and address than in single-block development. Causal paths of the drivers and constraints on project performance and progress bottlenecks pass through multiple types of resources, development processes, and move across both development phases and development blocks. These causal paths vary widely for different performance measures. They also change as projects evolve. This makes the drivers of and constraints on spiral acquisition project performance more difficult to identify than in single-block development projects. Progress bottlenecks can cause counterintuitive behavior, such as reductions in project cost by adding resources at a bottleneck. Understanding and exploiting the opportunities provided by these behaviors requires a deep understanding of the project structures and dynamic interactions that drive and constrain progress. Our modeling results indicate that spiral development is a significantly different approach to acquisition than single-block development, and requires different planning, resourcing, and management.

Observations and Assessments

Evolutionary acquisition seeks to spread out the technical risk over more development and process time via incrementing. We have shown with simulation that this can potentially improve risk management performance initially, but with higher overall costs and longer subsequent development durations. Our computational modeling indicates that incremental development costs more and requires more time to provide the same requirements than single step development. With regard to project risk, the increased complexity in a project using an incremental or spiral approach makes the isolation and effective management of progress bottlenecks more difficult than in single-step development.

The policy change is that spiral development now includes undefinitized increments and prescribes incremental development instead of single step development. All amorphous spirals will eventually become defined increments – in effect mini-programs. In years past they have often been implemented as sequential, separate, and successive product upgrades (such as the CH-47, UH-60, C-130, B-52 program examples). But current policy expresses these as more concurrent, frequent and continuous. Such concurrency adds complexity to development models, with attendant risks of over allocation of work, noise, error, duplication, and other inefficiencies from work deferral and divided effort in project management organizations. Additional oversight, reviews, contracting, testing, etc. will also likely affect transaction costs. If all requirements are known and an incremental approach is used, then there is a deliberate deferral of work to later increments and there will be a resultant increase in total development costs and durations for these same reasons.

Recommendations for Practice

1. Project managers need to be aware of the inherent risks of spiral development and take necessary precautions to balance those risks. Many tools and control measures are currently developed and available to assist project managers in balancing the risks of spiral development, such as technology readiness levels, configuration

management, technology performance management, real options, project phasing, risk management, earned value management and organizational design.

2. Incremental and spiral development projects provide additional *opportunities* for managing development risks that are inherent in the project design. These include project planning decisions about the number and concurrency of development blocks, and the requirements and associated technologies and design components to be included in specific blocks. This planning provides opportunities to anticipate where critical progress bottlenecks may occur and design how to best monitor and respond to them.

3. Product attributes may help determine the suitability of spiral development. PMs should consider such characteristics as: mutability, time criticality, man-rating, modular interdependency, key parameters of capability versus range of requirement attainment (i.e. binary vs. continuous), and the relative amount of concurrency among increments.

4. Progress bottlenecks in incremental and spiral development often oscillate between process constraints (e.g. availability of work due to upstream progress) and resource constraints (e.g. developer or project management quantities or productivities). Successfully addressing a constraining progress bottleneck often shifts the progress constraint to a different location in the project. Therefore, a structured and interdisciplinary practice of identifying and addressing bottlenecks can improve performance.

5. Configuration management accountability must be assigned and kept to maintain supportability, failure mode identification and causality and prevent the variety generated by evolutionary acquisition from reducing total product performance.

Discussion

Boehm's latest book on software development advocates balancing disciplined (more rigid) and agile (more flexible) methods to capitalize on the benefits of both. Discipline is needed as a control mechanism to avoid risk, but agility is needed to respond quickly to customer needs. Saying, "One size fits all is a myth," he advocates a balanced approach based upon risk. Consistent with our findings, he also advocates the more disciplined, risk-averse approaches for projects that are mission/safety critical, larger in size, and have more stable requirements.

Although today's policy of evolutionary acquisition is prescribed as a development methodology, it is actually focused more upon what -- not how -- we develop. As such, it is about doable scope, reducing risk via exclusive use of mature technology. The Cost As an Independent Variable and other requirement-limiting initiatives (i.e. elimination of MILSPECs) were earlier attempts to accomplish this, by encouraging product performance trades to keep cost estimates fixed. Like CAIV, this likely means trading performance requirements for earliest deploying increments.

Conclusions

It could be summarized that spiral development was at its inception and is at its extension all about risk. Paradoxically, it is an agile method envisioned to reduce risk, and yet can potentially add its own. On the one hand, a spiral or incremental approach allays risk by reducing scope to render only the highest priority capabilities with the exclusive use of mature technology, and obtains early and continuous feedback from the environment for follow-on developments. On the other hand, it introduces concurrency during advanced development and adds variety in production, with all their attendant management challenges.

We've suggested that a one-size-fits-all methodology for DoD system development may not be appropriate, and have offered for consideration several product attributes that might help determine the applicability of the spiral approach. We speculate that spiral development may serve better than single step development for initial capability when products are mutable, time critical, non-maintenance intensive, and have continuous (vs. binary) or uncertain requirements, short cycle times (less knock-on effects), sequentially phased development, and modular independence. In contrast, spiral development may not be appropriate when there are safety or man-rating concerns and have attributes opposite to those above. In particular, program managers should understand the nature of their product requirements with regard to their range of attainment and relative to key parameters of capability, and vis-à-vis the readiness level of their enabling technologies. Some key features may indeed be binary, and others may have significant ramifications of partial attainment – such as propagated change across the entire product componentry (as in weight reduction), versus a more independent modular modification.

Open design standards will not always be incorporable, and product variety will emerge, with and without backward compatibility, interoperability, etc. Variety is both an asset (for end users) and a liability (for manufacturers, owners and supporters). As such, to compensate for product variety risk, we posit that acquirers must "own" the design and emphasize configuration management, keeping or assigning responsibility for that function, and maintaining *accountability* for it.

Our title – "from amorphous to defined" – alludes not only to *product specification,* but also to *risk realization* in spiral development. Spiral development has inherent challenges, both strategic and tactical, of which PMs must be aware. We've highlighted and illustrated them here, as well as showing that spiral development can indeed work – especially for technically mature and mutable products with open or elegant architecture. Program managers must be aware of these inherent risks, and take necessary precautions to balance them with tools that we have mentioned.

Finally, *stability* is the quest in all things programmatic – for funding, requirements, design, production configuration, etc. But in an unstable world, and with the future being necessarily uncertain, the tension between control and change is probably unending. PMs do have some tools for coping, and being forewarned is being forearmed. PMs are used to concurrency and change, as they are largely what make project management what it is – a balancing act. Mechanisms for control of risk include many well-known project management tools. Organizational and cultural factors such as leadership, trust and accountability play a significant role as well. Successful use of these tools to balance control and risk in projects with a high rate of change and concurrency is an area for our further research, in order to improve our understanding and use of evolutionary acquisition.

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction—The Inevitability of Change

We are told in Diogenes Laertius's *Lives and Opinions of Eminent Philosophers* (early 3rd century) that the Greek philosopher Heraclitus (c.535 - 475 BC) was the first to observe and say, “Everything flows; nothing stands still,”—the popular derivation of which is, “The only constant is change.” Indeed, everything does seem to change, evolve and give rise to variety in the world. Since his work in the 1830s, Charles Darwin receives much of the credit for furthering a theory of biological evolution. While not the first to have the idea, he associated observations of species *variety* on the island of Galapagos with species *environment*, and suggested that nature selected the variations that were the fittest (Darwin, 1859). In its time (and even since), the idea was considered radical and a threat to the religious and social order of things. Mere variety itself can be controversial, since, paradoxically, variety is appreciated in some domains (Cowper, 1731-1800)[1] and abhorred in others (Neave, 2000, March 2).[2] At the core of the subject of evolutionary acquisition are ideas and phenomena about variety and change. As a policy for system development, it is controversial too. As with Darwinian concepts, product evolution involves *information* transfer, interaction with the *environment* and unpredictability of change outcomes. But unlike evolutionary biology, product variations and selections occur frequently and are non-random. Much of what the authors have found in their following research on spiral development and project management is about how managers must cope with product variety and change. Using case study analyses, review of current subject literature, and computational modeling, the focus of our research was to ascertain the acquisition management implications of spiral development, obtain lessons learned in past programs as

---

[1] See also: Kerr (1979, p. 65) about the basic human need for variety and complexity. Ashby’s Law of Requisite Variety states that the internal regulatory mechanisms of a system must be as diverse as its environment in order to cope with the variety of challenges imposed by it (Ashby, 1960).

[2] “Variation is nasty: it makes things difficult, unpredictable, untrustworthy: bad quality.” “In a big way, bad quality means too much variation, good quality means little variation.”

applicable to future development efforts, model and simulate projects using different acquisition approaches, derive predictions and make recommendations to project managers for the effective and efficient harnessing and implementation of spiral development.

Projects have long been defined as *unique* and *temporary* enterprises, as opposed to common and ongoing operations. The latest (2004) version of the Project Management Body of Knowledge (PMBOK) increased its emphasis upon the term "*progressive elaboration*" to describe a third fundamental characteristic of all projects. It means, "developing in steps and continuing by increments; worked out with care and detail; developed thoroughly" (PMBOK, 2000; PMBOK, 2004, p. 6). This term relates to project uncertainty and describes the eventual realization of project scope only after multiple iterations of planning. The PMBOK asserts that progressive elaboration is both a necessary *characteristic* of projects (occurring throughout their lifecycles), as well as a *technique* for development of product specifications. It is accomplished via the learning that takes place over time as project ambiguity resolves, so that project scope becomes more explicit and detailed (as opposed to "requirements creep" which is considered uncontrolled change). The PMBOK later asserts that change in the course of projects and products is inevitable, and mandates the need for a *disciplined change-control process* to control its impacts—from inception to completion (PMBOK, 2004, p. 119).

New Terminology and a Mandate for Variety

The Department of Defense has also put into effect new terminology in recent years pertaining to project management and product development methodologies, with often vague or subtle differences in meaning from older terms. Examples are: phased acquisition, agile acquisition, iterative design, rapid prototyping, pre-planned product improvement (P3I), product-improvement program (PIP), evolutionary acquisition, spiral development, incremental development/capability, planned upgrades, and modernization through spares. Others have used related expressions such as Rational Unified Process Framework, successive limited comparisons, and even “muddling through” (Sylvester & Ferrara, 2003)[3]

In the year 2000, the Defense Department promulgated the term “evolutionary acquisition” (EA) in its policy documents governing the strategy for acquisition of materiel, and mandated such strategies as the preferred approaches (USD(AT&L), 2000, October 23). Later elaborated as spiral and incremental strategies, these approaches contrast in principle to others that utilize more serial, sequential or singular efforts to arrive at a product solution (though not necessarily precluding the use of iterative planning/designing processes). They are often termed as: single-step-to-full-capability, grand design, big bang, technological leap, waterfall, rational-comprehensive, and the unified development method (Mooz, Forsberg, & Cotterman, 2005, p. 354). The overarching goals and principles of the DoD’s evolutionary acquisition were explained as follows:

> To ensure that the Defense Acquisition System provides useful military capability to the operational user ***as rapidly as possible***, evolutionary acquisition strategies shall be the preferred approach to satisfying operational

[3] Even social scientists have espoused the advantages of incremental progress in decision-making such as in Lindblom’s famous 1959 public administration classic, The science of muddling through: Lindblom, C. E. (1959). *Public Administration Review*, 19 (Spring), (Reprinted (1977). In F. A. Kramer (Ed.), *Perspectives on Public Bureaucracy* (2nd ed.) (pp. 132-150). Cambridge, Massachusetts: Winthrop Publishers).

needs. Evolutionary acquisition strategies define, develop, and produce/deploy an initial, militarily useful capability ("Block I") based on proven technology, time-phased requirements, projected threat assessments, and demonstrated manufacturing capabilities, and plan for subsequent development and production/deployment of increments beyond the initial capability over time (Blocks II, III, and beyond). (USD(AT&L), 2000, October 23; emphasis added)

See Figure 1.

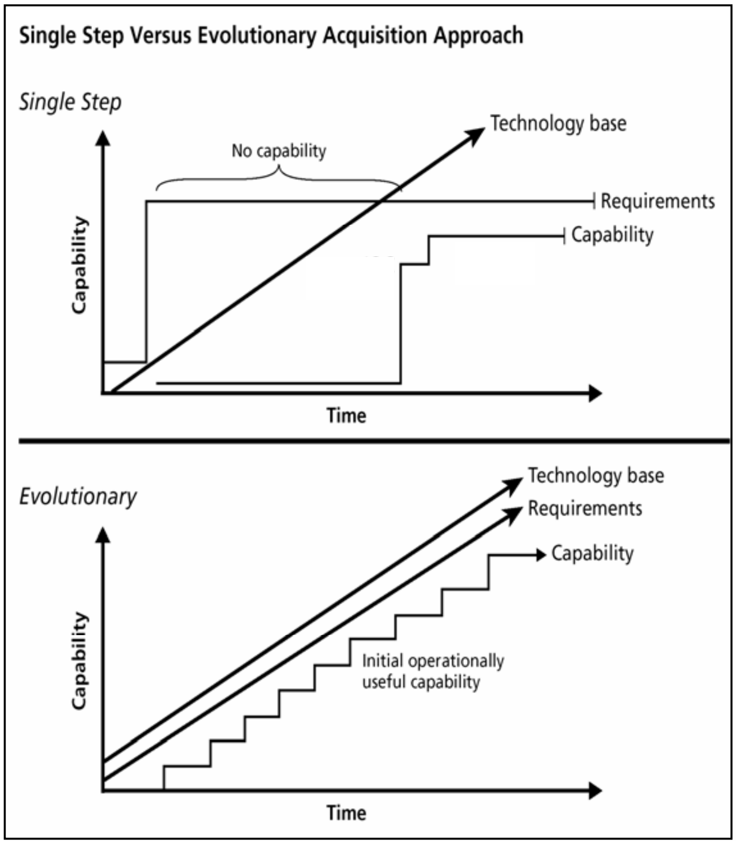


Figure 1. Incremental Capabilities (adapted from Lumb, 2004)

The DoD later defined an “increment” the following way:

An increment is a militarily useful and supportable operational capability that can be effectively developed, produced or acquired, deployed and sustained. Each increment of capability will have its own set of attributes and associated performance values with thresholds and objectives established by the sponsor with input from the user. (Chairman of the Joint Chiefs of Staff, 2003, June 24)

Initially, however, the DoD's definitions of spiral development were imprecise, and were exceedingly so for the next two years. "Spiral development" had been used since 1985 in the software community, coined by Dr. Barry W. Boehm, Chief Scientist of TRW's Defense Systems Group (Boehm, 1985, pp. 22-42). He also served from 1989-1992 as the DoD's Director of the DARPA Information Science and Technology Office, and as Director of the DDR&E Software and Computer Technology Office. When "spiral development" was rolled out by the DoD in 2000, it was first described as *a development process within product increments*, for example:

> Spiral Development is an iterative process for developing a defined set of capabilities *within one increment*. Each increment will include multiple spirals. This provides interaction among user, tester, and developer throughout system development. In each spiral, requirements are refined and allocated to the design. Then coding, fabricating, and integration is accomplished, either physically or via modeling. The system or model is then tested and results assessed. The learning from this spiral is then applied to the next spiral. This process is repeated until we have fully developed a system *concept*, then a development *baseline*, and finally, a *capability* that meets warfighter needs. (AFIT, 2007; Hawthorne & Lush, 2002, August)

Boehm's earlier work had pointed out that not only could software developers demonstrate functionality in an incremental way, but management could also commit corporate resources in an incremental way. But "rapid" and "evolution" are terms that don't go effectively together. And confusion continued in the acquisition community throughout 2003—when definitions emerged in midyear and were published in the next revision of *DoDI 5000.2* in an attempt to clarify the difference between spiral and incremental development as similar but different *processes within an evolutionary acquisition strategy (*Washington Technology):

> 3.3.2. The approaches to achieve evolutionary acquisition require collaboration between the user, tester, and developer. They include:
>
> 3.3.2.1. Spiral Development. In this process, a desired capability is identified, but the *end-state requirements are not known* at program initiation.

Those requirements are refined through demonstration and risk management; there is continuous user feedback; and each increment provides the user the best possible capability. The requirements for future increments depend on feedback from users and technology maturation.

3.3.2.2. Incremental Development. In this process, a desired capability is identified, an *end-state requirement is known*, and that requirement is met over time by developing several increments, each dependent on available mature technology. (USD(AT&L), 2003b, May 12)

Furthermore, of the two approaches to evolutionary acquisition strategy, spiral development was declared the preferred process for execution (USD(AT&L), 2003a, May 12). In 2003, the Congress sought to define these terms as well, perhaps so that completely new development efforts or programs could not be disguised as incremental spirals or product improvements.

(g) Definitions.- In this section: “(1) The term ‘spiral development program’, with respect to a research and development program, means a program that - “(A) is conducted in discrete phases or blocks, each of which will result in the development of fieldable prototypes; and “(B) will not proceed into acquisition until specific performance parameters, including measurable exit criteria, have been met. (US Code, Title 10, 2002)

For the acquisition workforce today, some confusion still persists with the DoD’s terminology, and certainly with the broader implications of the policy and its tactical implementation (Lorell, Lowell, & Younossi, 2006). To fully differentiate between old and new terminology and process criteria, the instructional and leadership arms of USD (AT&L) distributed the table below (Figure 2) in several presentations during 2003-2004 (Bruns, 2003, July 30).

Development Strategy Comparison Table

Criteria / Acq Strategy or Dev Process	Single Step to Full Capability	Pre-planned Product Improvement (P³I)	Evolutionary Acquisition	
			Incremental Development	Spiral Development
Full requirements defined at outset	Yes	Yes	Yes	No
Useful intermediate capabilities	No	Yes	Yes	Yes
Multiple iterations	No	No	Yes	Yes
All capabilities required in initial increment	Yes	No	No	No
User feedback from earlier iterations used to define final requirement	No	No	Yes	Yes
Other characteristics	Used as the traditional acquisition strategy	Achieves increased capability from maturing technology with architecture in place	Developmental process when full requirements defined at outset	Developmental process when full requirements not defined at outset

Figure 2. Development Strategy Comparison Table

As illustrated, what this all means in the simplest of terms is that we now have a mandate for *all programs to have multiple product releases*, some of which will have defined requirements while others are more amorphous. For the incremental development approach, this involves the deliberate deferral of work to a later project phase. Future adaptability is an inherent development objective for spiral and incremental approaches. However, if we look at programs over their extended lifecycles, it could be argued that many, if not all of them, have all been developed with (initially unplanned) continual spirals or increments of refreshment and improvement (such as the CH-47, UH-60, C-130, B-52 aircraft programs).

THIS PAGE INTENTIONALLY LEFT BLANK

Reducing Cycle-time and a Move toward Evolutionary Requirements

The policy for evolutionary acquisition strategy was aimed at improving all parameters of program success, but clearly and explicitly, its single most important objective was to *reduce long product cycle-times* to deliver operationally useful equipment. The attainment of agility or flexibility in what had been a very rigid requirements process was an implicit objective within the concept, but was nonetheless important from an implementation perspective. Commensurately, the Joint Capabilities Integration and Development System (JCIDS) process was changing in parallel with the *DoD 5000* series revisions and appeared in five different editions between August 1999 and May 2005. As one of its principal modifications, it prescribed a series of three evolving requirements documents to describe attainable capabilities from initial conception, through design, to production (Chairman of the Joint Chiefs of Staff, 2005, May 11).[4]

As previously mentioned, project management differs from operations management in that all projects are unique and exist for a limited amount of time, and with significant uncertainty. Uncertain events or conditions that can negatively affect project objectives operationally define risk (PMBOK, 2004, p. 5). Activity concurrency is a necessary aspect of projects for efficiency of execution, but only to the extent that the total scope is accomplishable. Otherwise, technical performance risks, as well as schedule and cost risks, emerge. Like others who operate in a strategic realm, project managers see themselves within an environment of volatility, uncertainty, complexity and ambiguity. Nevertheless, they are expected to predict project outcomes in terms of cost, schedule and performance. Project risk (typically schedule, budget and technical performance risk) is often viewed in terms of

[4] These requirements documents are the Initial Capabilities Document (ICD), Capability Development Document (CDD) and Capability Production Document (CPD), approved in support of Milestones A, B, and C respectively.

adequacy of available information about the project environment and the potential effects of management actions (Pich, Loch, & De Meyer, 2002, August 8). Large defense systems are very complex, consisting of diverse hardware and software sub-systems, multiple suppliers, numerous interfaces, etc. Worse, the current environment of rapid technological change has become particularly problematic for such projects with long product cycles. Because of this "churn," it is proving more and more difficult to fully define technical specifications—or even the complete set of system functional characteristics and operational capabilities—before commencing advanced development. Project uncertainty and risk seem to be increasing.

Earlier (1990s-era), DoD acquisition reform initiatives took aim at such ambiguity and uncertainty and sought purposefully to reduce the product cycle by alleviating the information gap and technology lag via measures such as: alpha contracting, advanced concept technology demonstrations, pursuit of commercial-off-the-shelf products, and Integrated Product and Process Development (IPPD).[5] During this era, it was thought that insufficient involvement of numerous and diverse project stakeholders, particularly early in the program's life, led to project changes later on that were more costly to implement. IPPD was adopted as a management process (Perry, 1995, May 10) to encourage cross-functional teamwork and promote *collective wisdom*. Employment of IPPD was specifically to facilitate meeting cost and performance objectives, as well as to field products sooner, via the continuous collaboration within Integrated Product Teams (IPT). But in the main, it was about early and complete *requirements capture* [6] through collaboration.

As concerns over DoD acquisition program costs and cycle-times continue in the current mid-2000s era, the DoD has not abandoned the use of IPPD. But by

---

[5] Of 63 named 1990s-era acquisition reform initiatives, many sought to reduce bureaucracy, modernize and streamline processes, and reap a resultant reduction in overall cycle-time. However, these four as mentioned appear directly oriented against technology uncertainty and inadequate information.

[6] See Bruce, M. & Cooper, R. (2000). *Creative product design.* West Sussex, England: Wiley & Sons, for an extended coverage of requirements capture management.

embracing evolutionary requirements and acquisition, it has acknowledged that information will never be complete, either from stakeholders or with regard to ever-changing technology. It now implicitly concedes that developers will learn about their design over time (“requirements realization”), and users will accretively gain knowledge about how they can better use the new capability (“product discovery”).[7] Thus, a major paradigm shift for product development has occurred in the DoD: *from a collaborative quest to capture and address all requirements early on, to an allowance of eventual requirements discovery with full attainment only after visualization, feedback and environmental changes occur along the way*.

[7] The authors’ terminology for what has so often been observed from their experiences. Most of us have long known that full realization of requirements and visualization of the product often takes multiple iterations of design, with feedback loops from modeling and testing activities. And sometimes the customer doesn’t fully realize what can be done with the product until it is in hand. We call that *product discovery*, and the authors can cite several examples of this in both commercial and defense applications (i.e., cell phones as improvised explosive device triggers, etc.).

THIS PAGE INTENTIONALLY LEFT BLANK

The Enabler: Mature Technology Reduces Risk

This is not to say, however, that the DoD has in its policy embraced technological uncertainty for the commencement of advanced development. Quite the contrary—for at the very heart of the evolutionary acquisition strategy is the requirement for the exclusive use of mature technology to reduce project risk. The impetus for this undoubtedly lies in the body of work by the Government Accountability Office (GAO) over the last ten years,[8] which has obviously and greatly influenced the *DoD 5000* series. The GAO encourages the use of knowledge-based processes and specifically separates technology development from product development. It characterizes three *knowledge points* in the course of product development as:

Knowledge Point 1	Matching of resources and needs (time, funding, technology, and requirements)—at the point of program initiation (corresponding to DoD's Milestone B).
Knowledge Point 2	Stable product design (typified by having 90% of component drawings complete)—midway through advanced development (DoD's System Development and Demonstration Phase) at the point of system-level critical design review (corresponding to the DoD's Design Readiness Review).
Knowledge Point 3	Mature production processes: proven products with all key manufacturing processes in statistical control and meeting cost, schedule, and quality targets. Described by the GAO as the start of production (corresponding to the DoD's Milestone C—though some might argue that such knowledge is not completely realized until Full Rate Production[9]).

---

[8] See in particular: GAO/NSIAD-98-56; GAO/T-NSIAD-98-123; GAO/NSIAD-99-162; GAO/T-NSIAD-99-116; GAO/T-NSIAD-00-137; GAO-01-288; GAO-02-701; GAO-03-57; GAO-04-53.

[9] Statement by US Army Colonel (Ret.) Mike Boudreau, former PM for the Family of Tactical Wheeled Vehicles (FMTV), in correspondence with GAO authors, May 19, 2006.

The GAO has claimed that genuine assessments of program knowledge acquired at these control points will reveal whether the programs and their requisite investments should proceed or be halted. They argue that shorter product cycle-times are the hallmark of program success and, therefore, should be limited to five years for more frequent introduction of new technologies into weapon systems, speeding them to the warfighter. We note that this is not much longer than the average development time for a new model of automobile—typically 3-4 years—which occurs in a very mature and cyclical industry (Kim, 2002, June). This target may ignore the significantly greater amount of technology development required in many DoD projects compared with most automobile development projects.

Most emphasized by the GAO (in the many reports reviewed by these authors) is the aspect of technology maturity before commencement of advanced development. The Office applies a 1-through-9 rating scale of *technology readiness levels* (TRL) that was developed by the National Aeronautics and Space Administration, adopted by Army and Air Force research laboratories, and recently implemented in the *DoD 5000* series (in particular, the *Defense Acquisition Guidebook*—formerly *DoD 5000.2R*). Until recently, the DoD had no specific requirements for use of TRLs, but levels 6 and 7 now satisfy its guidelines for technology maturity at Milestone B. TRL 6 states that the technology has been demonstrated in a *relevant* environment (simulating the key aspects of the operational environment), and TRL 7 is its demonstration in an *operational* environment (that which addresses all operational requirements and specifications required of the final system, to include platform/packaging). The GAO clearly prefers TRL 7 as the level of technology maturity that will represent a low and satisfactory risk for starting product development (GAO, 2005, November 15). The Office acknowledges that users may not initially receive the *ultimate* capability under this approach, but that the initial capability will arrive predictably *sooner and cheaper* (GAO, 2005, November 15). (See Figure 3.)

Technology readiness level	Description
1. Basic principles observed and reported.	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples might include paper studies of a technology's basic properties.
2. Technology concept and/or application formulated.	Invention begins. Once basic principles are observed, practical applications can be invented. The application is speculative and there is no proof or detailed analysis to support the assumption. Examples are still limited to paper studies.
3. Analytical and experimental critical function and/or characteristic proof of concept.	Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.
4. Component and/or breadboard validation in laboratory environment.	Basic technological components are integrated to establish that the pieces will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of "ad hoc" hardware in a laboratory.
5. Component and/or breadboard validation in relevant environment.	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so that the technology can be tested in a simulated environment. Examples include "high fidelity" laboratory integration of components.
6. System/subsystem model or prototype demonstration in a relevant environment.	Representative model or prototype system, which is well beyond the breadboard tested for TRL 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high fidelity laboratory environment or in simulated operational environment.
7. System prototype demonstration in an operational environment.	Prototype near or at planned operational system. Represents a major step up from TRL 6, requiring the demonstration of an actual system prototype in an operational environment, such as in an aircraft, vehicle or space. Examples include testing the prototype in a test bed aircraft.
8. Actual system completed and "flight qualified" through test and demonstration.	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications.
9. Actual system "flight proven" through successful mission operations.	Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. In almost all cases, this is the end of the last "bug fixing" aspects of true system development. Examples include using the system under operational mission conditions.

Figure 3. DoD Technology Readiness Levels (GAO, 1999, July 30)

In some respects, developing only mature technology as a fundamental program requirement is similar to an earlier attempt to constrain project scope. Cost As an Independent Variable (CAIV) was an acquisition reform initiative that emerged in 1995 as a means of trading scope, or system performance, to achieve cost objectives. It was one of very few initiatives that were oriented on *what*, not *how* (i.e., processes) the DoD acquires its materiel.[10] To date, its actual savings benefit has been difficult to quantify, and qualitative measures have shown mixed results

[10] Some may also assert that the moratorium against MILSPECS was similar in its thrust to reduce unnecessary scope of work, but we believe many specifications to be as much prescriptive (i.e., "how") as they are descriptive.

(RAND, MG291, 2005). The practice of using requirement attainment objectives and thresholds was another way to facilitate performance trades for cost.

When fully realized, *it is the exclusive use of mature technology in system development programs that is the key enabler of evolutionary acquisition* strategy, facilitating the rapid transformation of applied technology to end-item capability. Thus, it is the third of three principal observations, all of which are paradigm shifts, that we have recently observed: (1) that the DoD would now mandate program strategies for all programs to have multiple product releases of the same item, (2) that requirements would be deferred or allowed to evolve over time, and (3) that high levels of technological maturity would be requisite for commencement of advanced development, with an intended reduction of technical risk (and thus, project schedule) (USD(AT&L), 2003a, May 12, Enclosure—Additional Policy E1.14).

Policy Concerns

But there are questions and concerns about these major shifts that several authors have raised. While still a relatively new policy, observations and realizations about the outcomes of evolutionary acquisition and spiral development are only just beginning to emerge, until at least several major programs go through their entire lifecycle in this way. One of the first histories and analytical descriptions of evolutionary acquisition policy formulation came from Sylvester and Ferrara, in their 2003 discourse on *Conflict and Ambiguity Implementing Evolutionary Acquisition* (Sylvester & Ferrera, 2003). This piece gave some insight into the challenges and obstacles of evolutionary acquisition implementation—not from program office level—but from the perspective of strategic policy makers and subscribers at the Office of the Secretary of Defense (OSD) level, during their struggle to adopt the policy. In short, the authors explained the aforementioned confusion and ambiguity of the policy as it evolved from 1983 toward final promulgation in 2000, and then described the conflict areas caused by shifts in power among the organizational fiefdoms in the OSD and other affected institutions (i.e., Congress and the defense industry). In particular, they exposed the following major stakeholder communities and their respective *areas of concern* about evolutionary acquisition:

Congress	loss of control over DoD programs via specific and informed authorization and approval; the inability to keep the DoD accountable; unknown implications of requirements and budget flexibility required for evolutionary acquisition.
Military Departments	need to protect own acquisition programs and share of the DoD budget; retention of funding for follow-on capability increments; increased oversight; downstream logistics of multi-configuration products.
Defense Industry	disruptions to commercial processes and traditional approaches to business; competition for follow-on increments; lower-rate production runs after shorter R&D efforts.

Comptroller	controlling programs and holding them accountable; unknown implications of requirements and budget flexibility required for EA (program and budget "gaming" by services); "full funding" policy[11] versus open-ended requirements and fund streams.
Requirements/Users	sub-optimum capability; priority of what is needed versus what is currently attainable; loss of follow-on increments.
Test and Evaluation	loss of discipline and assurance of operational effectiveness & suitability; lack of comprehensive testing before several low-rate production configurations are released.

Sylvester and Ferrara's list of these policy concerns was not meant to be all-inclusive, nor does it take into account other communities, like program managers and logisticians, who also have issues about evolutionary acquisition implementation. But their essay about strategic conflicts within the emergent policy does provide valuable insight into some of the challenges of effective tactical implementation.

[11] The authors explain the dual meanings of this term later in this discussion.

Implementation Concerns

The authors of this discussion have also been attentive during the policy's turbulent progression. As researchers and former practitioners, we've had our own concerns about spiral development from both strategic as well as tactical standpoints, and with regard for its efficiency and effectiveness in application:

- We previously noted the *significant increase in OSD-level program decision reviews* under the new acquisition framework (Dillard, 2005), and suggested such additional centralization of control might work counter to the stated policy's innovation-fostering goals. Reviews serve as control gates where decision makers can stop, extend or grant permission for projects to proceed into the next phase. Program reviews, major-milestone or otherwise, at the OSD level have a significant impact on program offices as *off-core activities*. Much documentation must be prepared and many preparatory meetings are conducted enroute to the ultimate review. And while non-milestone reviews are generally considered to require less preparation effort, a considerable amount of effort managing the decision process is still expended. The latest *DoDI 5000.2* prescribes that, "In an evolutionary acquisition program, the development of each increment shall begin with a Milestone B, and production resulting from that increment shall begin with a Milestone C" (USD(AT&L), 2003b). Thus, program managers can expect to undergo the reviews determined appropriate for the initial increment of development in their program, as well as reviews specified for all of the follow-on increments. And our concern follows that the added time and effort expended on such control activities and added transaction costs might actually delay the arrival of capability (Franck, Melese, & Dillard, 2006) (see Figure 4).

- There will likely be *significant organizational impacts* of concurrent development and production within program management offices. Of concern is that the first increment's operational testing and production effort may now run parallel to the follow-on development effort for the next increment, presenting additional management challenges to the program manager. If designers are truly freed from development of the initial increment, they can be gainfully employed in the successive effort. But, if system components need to be re-worked as a result of incomplete realization of requirements or incomplete implementation of the technology in the first increment, there will be organizational stress and division of effort from the added concurrency. In either case, there will likely have to be duplicative or additional management elements employed in the organization as it is executing production and development at the same time. It would indeed be an unfortunate consequence to have two increments to achieve one set of capabilities take longer and cost more than it would have in a project structured to just one increment (Dillard, 2005) (see Figure 4). It is these phenomena that we have modeled with computational organizational design tools, which will be discussed later.

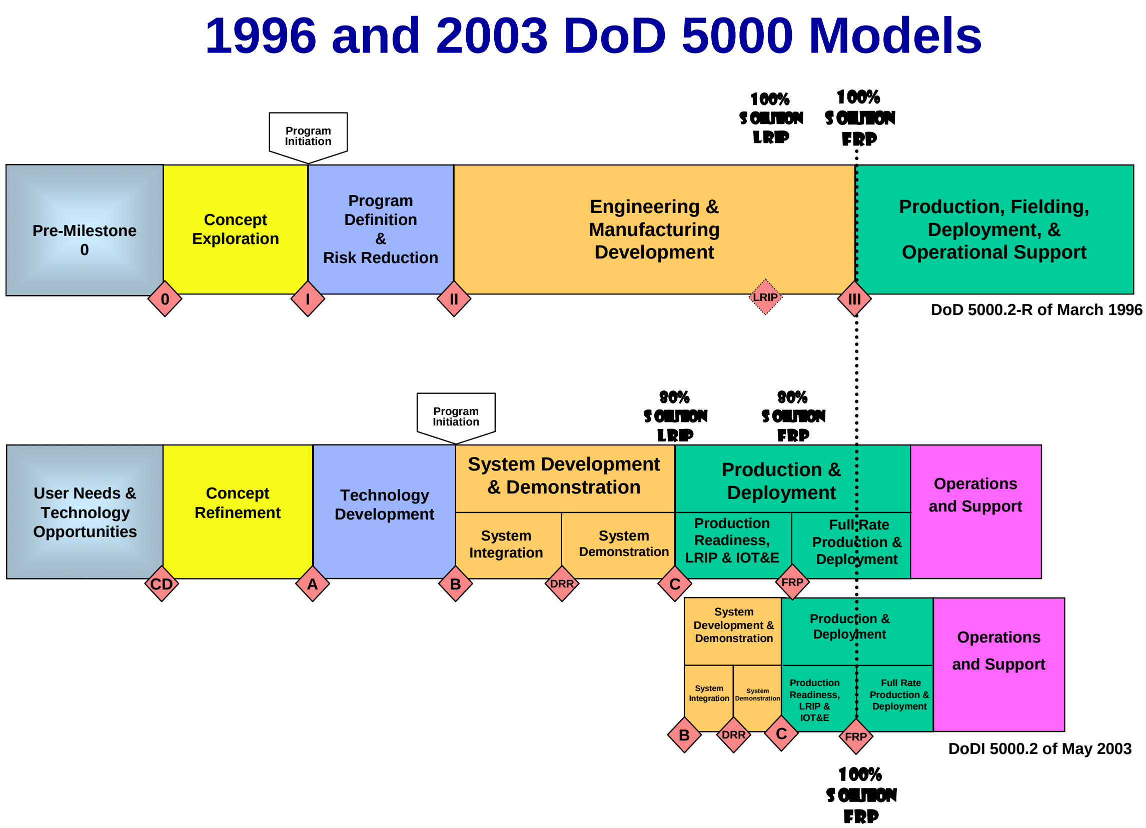


Figure 4. Comparison of 1996 and 2003 Acquisition Framework Models

- The GAO compiled a large body of convincing evidence that technology maturation efforts during advanced development have lengthened programs, with a resultant delay in capability arrival and substantial cost growth. Under the new framework, Milestone B is the formal declaration of program initiation and product (versus technology) development.[12] But, given the hypothetical *arrival* of technology maturity at some given point in time, it can be argued that the delay of program initiation until "the technology is demonstrated in a relevant environment"[13] can only come from more time spent in the preceding phases of Concept Refinement and Technology Development. Unless SDD (advanced development) is greatly shortened indeed, this could make less certain the potential of any real program cycle-time reduction, or worse—could *increase the likelihood of obsolete product technology* (or simply non-competitive state-of-the-art technology) at Milestone C (start of initial production).[14] (See Figure 5.)

[12] DoD policy greatly reflects the influence of the GAO Reports recommending increased product knowledge prior to business commitment. See GAO. (2002). Best practices—Capturing design and manufacturing knowledge early improves acquisition outcomes. 02-701. and GAO. (1999, July). Better management of technology development can improve weapon system outcomes. NSIAD-99-162.

[13] Which relates to Technology Readiness Level 6—Now in statute: amended in 2006, Title 10, United States Code Chapter 139 Sec. 2366a. Major defense acquisition programs: certification required before Milestone B or Key Decision Point B approval`(a) Certification—A major defense acquisition program may not receive Milestone B approval, or Key Decision Point B approval in the case of a space program, until the milestone decision authority certifies that—`(1) the technology in the program has been demonstrated in a relevant environment.

[14] Some seasoned program managers interviewed have seen technology languish in the laboratories, sometimes never transferring to system application—the fear being now that too much time will be spent in technology development with ineffectual efforts to "pull" from the technology base, versus driving or "pushing" the technology to maturity in the system-development phase.

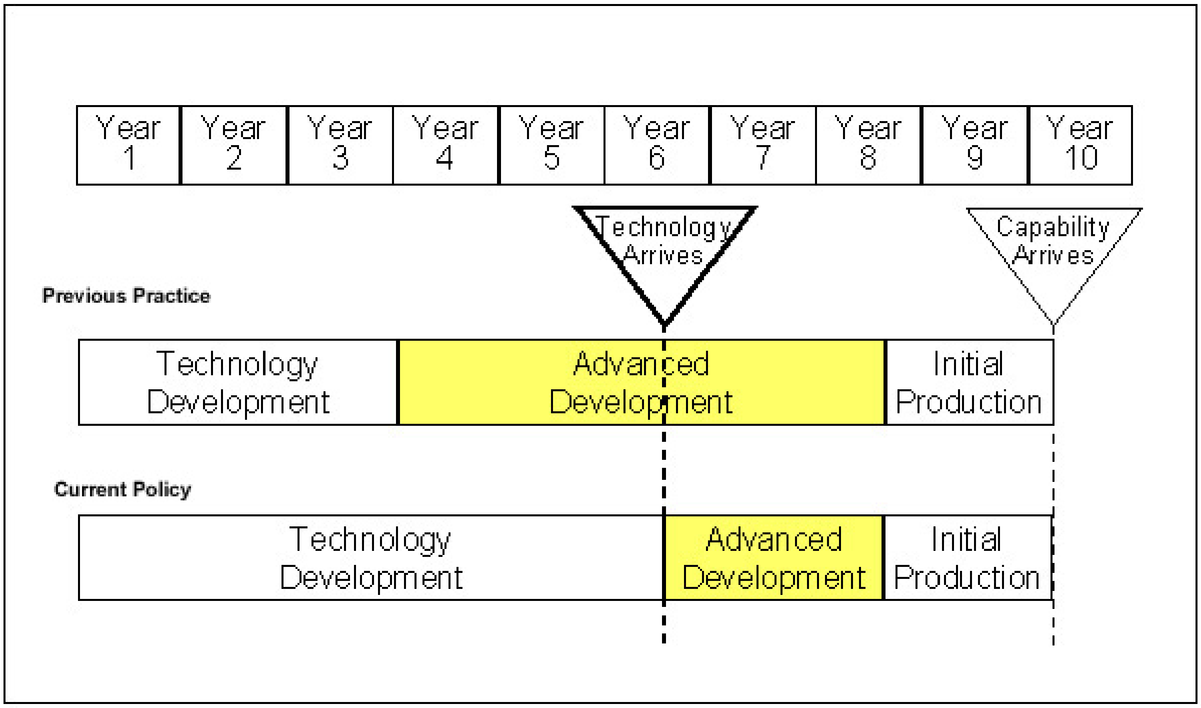


Figure 5. Lengths of Development Phases Relative to Technology and Capability Arrival

- The long held term, "full funding," pertaining to the total cost associated with an authorized quantity of militarily usable end-items for procurement within the fiscal year, was recently given an added meaning. Current DoD policy requires *full funding* for programs at Milestone B. In this sense, full funding also means having an approved current (and projected) resource stream with which to execute the program; i.e., program funding is included both in the budget and in the out-years of the FYDP *sufficient to cover the current and future efforts described in the acquisition strategy* (DAU, 2001). Expansion of the term was to ensure that programs would be less likely to exceed program estimates if they were not initiated until all forecasted resources were in place (USD(AT&L), 2003b).[15] However, evolutionary acquisition allows for one of two development processes to be implemented: (a) Incremental Development—in which end-state requirements are known, and will be met over time in several increments, and (b) Spiral Development—in which desired capability is

[15] DoDI 5000.2 states that: "Transition into SDD requires full funding (i.e., inclusion of the dollars and manpower needed for all current and future efforts to carry out the acquisition strategy in the budget and out-year program…)."

identified, but end-state requirements are not known at program initiation, and requirements for future increments are dependent upon technology maturation and user feedback from initial increments. *A special challenge is presented for obtaining realistic full-funding estimates for programs with uncertain requirements and numbers of increments.* Unplanned work is inestimable. Likewise, timing the arrival of RDT&E or Production funding via the Planning, Programming, Budgeting and Execution (PPBE) process for unanticipated discoveries that might suddenly emerge is an additional challenge for this calendar-driven and lethargic decision-support system. Much depends here upon the relatively *successive* or *concurrent* phasing of the follow-on increments. Where increments are defined, other financial and political aspects may also come into play, such as *maintaining the priority of funding for the successive increments*. (Since all programs compete for funding within the DoD budgeting process, division of programs into discrete increments would seem to make decrementing easier, if not more likely.)

THIS PAGE INTENTIONALLY LEFT BLANK

The Costs and Benefits of Variety

Evolutionary acquisition methodologies, in addition to potentially adding more concurrency during development, increase variety in production. Variety can be both a liability and an asset. Much has already been written about the obvious logistical challenges and ownership costs that can arise from having multiple configurations of deployed hardware end-items (Apte, 2005, June 30). Use of standardized or common components requires fewer inventories and a resultant cost savings, depending upon the need for maintenance and spares support (Ravindran, Phillips, & Solberg, 1987, p. 329). RAND's study of support considerations for the current mixed configuration fleet of Unmanned Aerial Vehicles (UAV) said, "Multiple aircraft configurations drive *multiple spare component packages* and, in the most extreme cases, may drive *multiple pieces of test equipment*, all significantly increasing long-term support costs" (Shaver, Lynch, Amouzegar, & Snyder, 2005; emphasis added). And changing production configurations is not viewed as efficient—due to supportability issues (regarding spares and maintenance) with lot, model, and type diversity. Reliability issues can also emerge because of insufficient testing of the changes. Depending on the degree of change, system validation and qualification become a concern if changes are not under strict control. And there may be *backward compatibility* and *interoperability* issues as well. Another burden is the training impact of mixed capabilities within the force or even within the same owning and operating unit.

In production—and for hardware in particular—a *stable design* is often the quest: to reduce *unwanted variation* and the potential for detrimental and unintended consequences. It is not that change or variety itself is deleterious, but we fear the penalties of unwanted change. Also, many project managers have long been taught to seek total requirements realization up front via rigorous IPPD and Systems Engineering Process (SEP) methodologies to avoid re-work, and because changing the design of a product later in its life (at least in the sense of performance enhancement or correction of flaws) is costly and inefficient. See Figure 6.

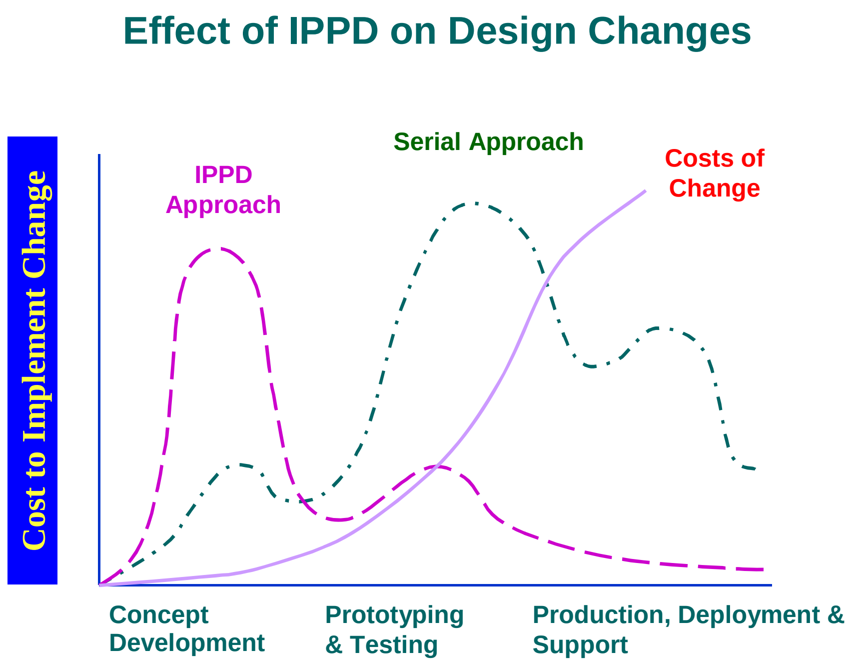


Figure 6. A Concept of the Relative Cost of Design Changes over Time

Still, design changes often seem to abound once a product is in production—where efficiencies can be discovered via learning-curve effects, and *minor engineering changes* can be applied for value. Continuous Production Verification Testing (PVT), and even Follow-on Operational Test and Evaluation (FOT&E), is conducted as deemed necessary to re-prove the system and allay the risks of unintended change propagation. Then may come the question of whether or not to retrofit previously manufactured items (to level the capabilities across the item population), and to what extent the items to be modified will become similar to the newer items produced.

Aside from ownership, the risks and costs of variety also come into play at the manufacturer's facility, with product-design changes cascading through manufacturing process design to manufacturing system consequences. Most recently, it has come to light that Airbus's 380 aircraft has been delayed for two years, costing perhaps $6 billion in profits, because of incompatibility between versions 4 and 5 of Dassault's same Catia computer-aided design (CAD) software (Duvall & Bartholomew, 2007). Production variety generates such expenses as: the maintenance of configuration documentation, testing, risk analysis, spares inventory, supply chain, and tooling. The new Ford Motor Company Chief Executive Officer, Alan R. Mulally, dramatically cut costs at his former job as president and chief executive officer of Boeing Commercial Airplanes by reducing the number of aircraft models from fourteen to four, and now purportedly plans to reduce Ford's eight brands as well (Langley, 2006, December). Variety equates to complexity for management, and it comes with a cost (as well as potential benefits for customers).

However, free markets appreciate variety in products and services. One MIT researcher asserts:

> Complexity is not an inherently bad property to us. Rather it is the coin of the realm in systems. You usually have to expend complexity dollars to achieve useful goals, such as increased functionality, efficiency or flexibility. (Moses, 2000)

Marketplace merchandisers provide variety for consumers who, on the whole, demand *selection* (points of product differentiation), and wish to benefit from the economic behaviors of competition. A mix of products is more likely to satisfy both mainstream and smaller niche needs. Most often, market needs and *annual business cycles* for revenue drive commercial decisions about time to product delivery—such as seen with the annual cycle of toys or automobiles. Commercial firms, then, are accustomed to making decisions about "doable scope" and are willing to defer offering product features that are less attainable (more risky) for the coming year's introduction to the market. But competitive threats against a new commercial market product launch do not typically involve loss of life or even livelihood.

It is along this vein that we take issue with some examples used by the GAO to provide rationale for DoD employment of evolutionary acquisition. Over the last decade, they have used products such as Maytag washing machines, commercial automobiles (the Jaguar, Lincoln Navigator, and Plymouth Prowler), commercial aircraft (Boeing 737 and 777) and commercial shipping (Polar Tanker), etc., as exemplars to make the case for a array of practices that the DoD should employ—such as design trades for reliability and reduced operating costs, use of mature technology, and evolutionary acquisition.[16] For the most part, we regard these commercial products as relatively "low-tech" on a comparative scale of DoD system complexity and capability. But more importantly, the GAO ignores product variety from the vantage point of *owner* versus that of the *producer*. The DoD is quite unique in that it almost entirely outsources capital projects for exclusively internal use. Companies such as Boeing and Jaguar and Maytag do the opposite—they conduct internal projects for external users. The concept is an important one, we feel, because of the implications of *ownership*—especially with regard to product variety. And if the extremes of combat environments are added for consideration and comparison of such products, it becomes clear that the risks of added complexity increase gravely.

A more representative commercial archetype, if there really were one, would be a product such as those within the United Parcel Service's truck fleet—a product created specifically for the internal use of UPS and to its unique specifications.[17] With a fleet of now 80,000 diesel-powered vehicles, delivering some 13 million packages per day, UPS has continuously (since 1935) explored the potential of alternative fuels for reduction in pollutants and fuel economy. Its latest excursion was in 1996, to introduce a truck using Compressed Natural Gas (CNG) manufactured by

---

[16] see GAO Reports 99-162, 03-57, and 98-56.

[17] Indeed, the GAO did reference the FedEx truck fleet in one of the above reports with regard to design trades for reliability and lower ownership costs, but not for the introduction of product variety and system evolution.

Freightliner Custom Chassis and Cummins Engine Company. The vehicles were built in 1996 and tested from 1997 to 2001 with a limited deployment of 101 vehicles, and confined to a small geographical area—Hartford, CT. The CNG trucks had 75% lower emissions for carbon monoxide, 49% lower oxides of nitrogen, and 95% lower particulate matter than the diesel trucks of similar age. But the energy-equivalent fuel economy of the CNG trucks was 27% to 29% lower than that of the diesel trucks, and the maintenance costs were 29% higher. Citing larger infrastructural issues, the UPS *CNG Report* cited lack of publicly accessible CNG refueling stations as a nationwide issue that deters the further deployment of such vehicles, and suggested that more economic incentives (tax credits and exemptions, fuel discounts, etc.) were needed (Dept. of Energy, 2002, August). With only 1% of its truck fleet now using alternative fuels, UPS has no current plans to procure more CNG fleet vehicles, but continues to watch the development and economics of new alternative fuels technology. This short example only serves to point out that *unique* users of *unique* equipment have *unique* ownership and support requirements; and product variety is not without its consequences.

THIS PAGE INTENTIONALLY LEFT BLANK

Do Product Attributes Affect Spiral Applicability and Outcomes?

Spiral development as a universal, "*one-size-fits all*" strategy may not always be appropriate. In addition to strategic and tactical implications about spiral development that we have already mentioned, more operational questions have surfaced of late: such as, whether certain product characteristics might encourage or discourage the use of this development approach. As already described, spiral development was conceived for alleviation of software risk from ill-defined solutions and uncertain requirements. From the literature and cases we've examined, we offer other product attributes below for program managers' careful consideration when planning product capability increments.

Mutability

We question whether products with different attributes (e.g., hardware vs. software, buildings vs. electronics) may lend themselves more or less to the use of a spiral development approach. Perhaps the foremost reservation is the appropriateness of the spiral development process for all project sizes and product commodities in toto, and the application of the spiral process to hardware products versus Boehm's original and most relevant application of this development approach toward software.[18] It would also seem appropriate that some regard be given to the second- and third-order effects of evolutionary acquisition, like: training, supportability, failure causality, mixed-unit capability, funding decrements, decision reviews, organizational impacts of concurrent development and production efforts, etc., before its general application. Our research was in part to ascertain some of the product/project parameters that make sense for spiral development. Boehm himself

[18] And the authors will be quick to acknowledge that software is indeed a huge and growing part of hardware systems large and small. Still, the spiral development framework in current literature applies overwhelmingly to the realm of software, not hardware.

warned of “hazardously distinct” spiral model imitations, and in his own words described his vision of the spiral process:

> The spiral development model is a ***risk*-driven *process model generator***. It is used to guide multi-stakeholder concurrent engineering of software intensive systems. It has two main distinguishing features. One is a ***cyclic*** approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of ***anchor point milestones*** for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions. (Boehm & Hansen, 2000, February 9. emphasis added)

Clearly, the conceiver of this spiral notion was oriented upon amorphous requirements and continuous stakeholder inputs *for the alleviation of project risk* with a very mutable product (Reed, 2006, December 16). The nature of software being in the digital rather than physical realm, it is particularly conducive to rapid and successive revision and *nearly costless production*. And even Boehm encourages varying from the spiral model as needed and reverting to a sequential model *if requirements are well established* and the project less risky.

Multiple product increments do not often appear in large, *static, singular* projects such as bridges, highways, office buildings, or in other project areas that have *typically long lead times or product cycles*, such as feature-length films, pharmaceuticals, etc. These are what we call *nearly immutable* products and are much different than smaller projects (like small application software development) with much shorter development periods. However, as with almost everything engineered that we can observe in the physical world, even these things can evolve and change with additions, spin-offs, sequels (and prequels), expansions, etc. Expansion of the long-standing San Francisco Bay Bridge and the now well-known Pentagon Renovation Program (enduring the attacks of September 11th, 2001) are examples (see Figure 7.)

Figure 7. CALTRANS San Francisco Bay Bridge Expansion Project

Cycle-time and Phase Concurrency

Akin to relatively mutable or immutable products, we have observed the successive product upgrades visible in long-running aircraft programs (See UH-60 Blackhawk and C-130 Hercules chronologies in Appendix A and B respectively) in which there are periods of production configuration stability, followed by improvement efforts, followed by another stable use period. Cycle-time for the development of each increment, and the relatively *successive* or *concurrent* phasing of the follow-on increments, will have a definite impact on program structure, budgeting, project complexity, and organizational issues, etc. For reasons that we will bring forth in our section on the computational modeling of spiral development, we have concerns about the conceptualization of spiral development programs with continuous and highly concurrent phasing of development increments, such as in the doctrinal Figure 8 below.

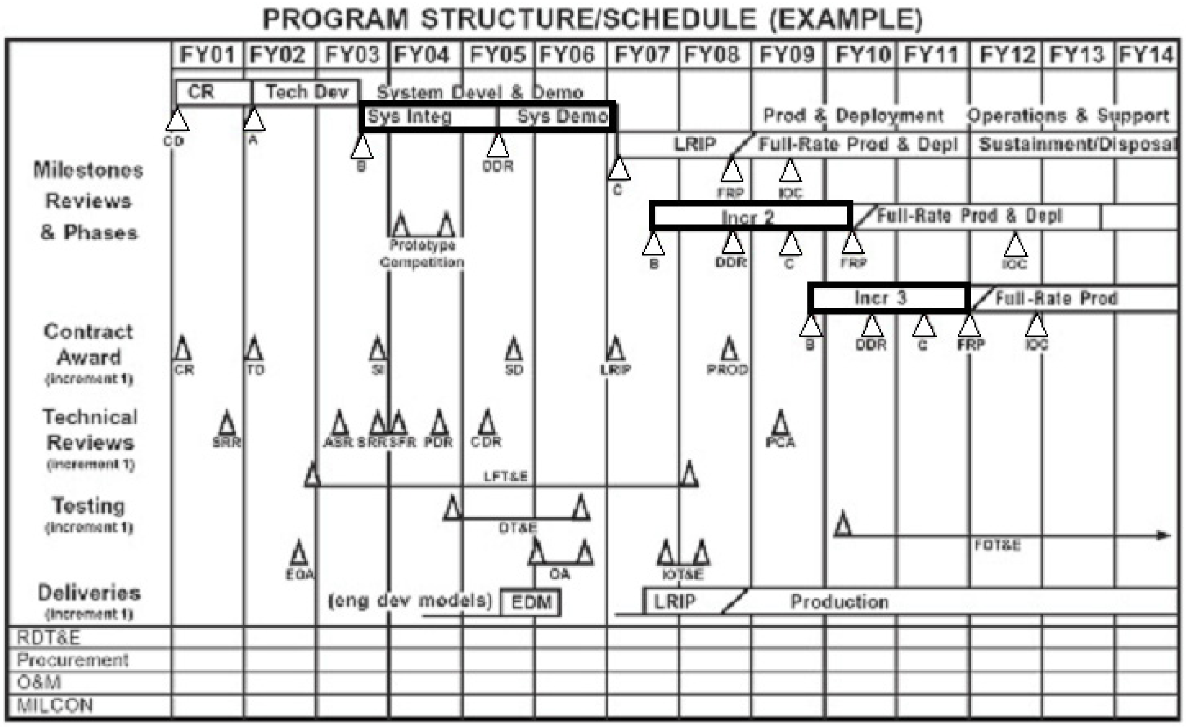
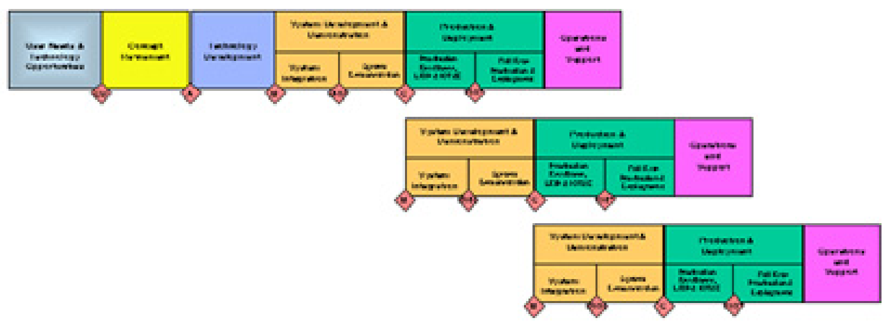


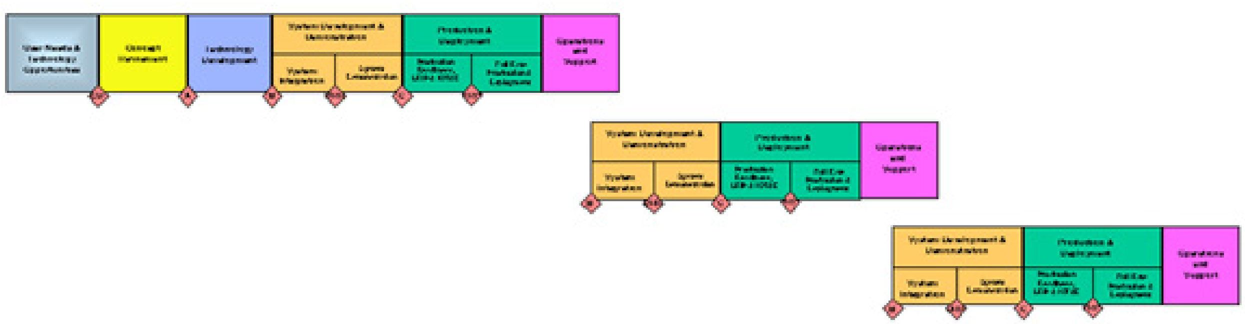
Figure 8. Example of Program Structure Showing Two Successive Development Increments (adapted from DAU, 2003, June)

We suggest that, though concurrency is a necessary ingredient for efficient project management, it has also long been correlated with risk (due to interdependence of activities), and might vary significantly with the types of activities underway (See Figure 9)—the inference being that periods of stable production configuration between development increments reduce complexity in program structure and attendant risks. Similarly, shorter cycle-times have less opportunity for knock-on effects or secondary consequences.

Particularly in matrix organization structures, as often the case with projects, there can be a tendency to staff multiple projects with a single specialist. The more projects a specialist supports, the less they are *proportionately* available to the projects due to "queuing inefficiencies." Availability decreases because of the need for *transition* between projects (physical, mental, learning curve, etc.). The end result has sometimes been shown to be large delays in project completion (Smith & Reinhartsen, 1998).Similarly, Ibrahim (2005) has shown that *discontinuous* *enterprise membership* is a contributing factor toward knowledge loss in organizations involved in large complex product development processes. Examining knowledge flows across product life cycles, members often are not engaged in all phases. Whether from rotation of duties or multi-tasking, a discontinuous member's inaccurate knowledge could cause a functional error at the individual level, which is not obvious at the enterprise's overall project level. Her findings support observations of knowledge loss continuing despite investments in information technology and knowledge management.



Development Spirals/Increments Concurrent with Initial Production



Development Spirals/Increments Concurrent with Later Production

Figure 9. Concurrency Relative to Types of Activity

User Risk (Safety and Time Criticality)

Time-critical or Enhanced Survivability Systems

We have discussed above the area of technological risk and the DoD's use of incremental or spiral approaches to resolve it (along with a compulsory policy for the advanced development of only relatively mature technology). But DoD products have expanded risk considerations beyond Boehm's models of commercial software. Extending the idea of project risk-as-a-driver down to the level of the end-user, it might seem logical to assume that *time criticality* of the need or mission, where risk of not achieving project success actually endangers customer lives, might be a significant factor in the appropriate application of the spiral process for reduced initial product cycle-time. Perhaps defensive systems are a good example. The immediate needs for a Rocket-Propelled Grenade (RPG) defeater or an Improvised Explosive Device (IED) neutralizer for currently deployed forces in Iraq and Afghanistan, for example, clearly dictate that lives will be lost if a near-term capability is not achieved. We also cite as an example the National Missile Defense (NMD) initiative, in which, in view of near-term threats, early deployment of even rudimentary capability has been deemed preferable to waiting for full capability. Such urgency likely precludes full and certain requirements specificity.

Man-rated Systems

In an almost opposite vein, *non-man-rated systems*, such as Unmanned Aerial Vehicles or cave-exploring robots—capabilities in which operator lives are not at risk if the product fails—may also lend themselves readily to rapid innovation and risk-less experimentation cycles.

However, user hazard levels for *man-rated systems* may be a different matter. Configuration variety adds technical complexity with sometimes unpredictable interactions. In such projects as pharmaceuticals, aviation, vehicular transportation, etc., producers mitigate safety risks with thorough analyses, documentation reviews, testing and other control and verification processes. By their very nature—with lethal hazards for the end-user, and typically lengthy approval requirements—these may not be good candidates for a spiral approach.

Production Quantity

Aside from software-versus-hardware mutability, requirements uncertainty, and time/life criticality, we questioned whether production quantity is an attribute that might also help determine whether a spiral approach is best. There seems to be a view, in addition to some risk factors mentioned above, that these might be driving factors for NASA's acquisition strategy determination. In June of 2006, the Center for Strategic and International Studies' Human Space Exploration Initiative and Defense Industrial Initiatives Group hosted a conference on *Spiral Development, Real Options, and Other Development Methodologies*. Its purpose was to explore these topics in an open workshop forum to gain programmatic and financial perspectives and search for tools to mitigate space-related technology development problems. These authors attended and made presentations about their previous acquisition research.

One panelist was Dr. Robie Samanta Roy**,** Assistant Director for Space Aeronautics from the President's Office of Science and Technology Policy (Roy, 2006, June), who formerly had worked with the Congressional Budget Office reviewing the *Aldridge Commission* (also known as *The President's Commission on Moon, Mars, and Beyond*) on how to implement the human space exploration vision laid out by President George W. Bush in January 2004. In his statements at the conference, he described spiral development as a "go-as-you-can-pay strategy," alluding to fiscal constraints and the incremental commitment of funds at decision points facilitated by the approach. However, for the development of the new Crew Exploration Vehicle (CEV), he suggested NASA was taking a different stance, perhaps because of *no mass-production of such systems* and the "front-loaded technology maturation" efforts peculiar to space systems acquisition. He stressed the need for clearly defined requirements for development of only a "handful" of space exploration vehicles and for primary focus to be upon *an architecture*.

Another panelist, Mr. Chris Scolese, NASA's Chief Engineer said regarding spiral development, "NASA's business is a little bit different—*We don't build lots of* *anything*," (implying that long production runs encourage the application of a spiral approach). Tacitly rejecting the spiral approach, he stressed the risk aspect of NASA missions in terms of project costs and human life (e.g., earth orbit versus deep space) and framed the use of real options as "trading risk, not ROI (return on investment), for value." Agreeing with Robie Samanta Roy, he said that the spiral process "will still be there" as NASA systems are "software intensive." But he also said, "No two identical spacecraft are the same," which seemed to contradict his idea that like configurations are a necessity among small production lot sizes.

Indeed, naval shipbuilders say the same thing about variation among individual ships, or within flights, of the same class. And even one-of-a-kind *nearly immutable* projects like skyscrapers and bridges can be later re-modeled and refitted, as we mentioned above.[19] It may instead be that NASA feels itself within a financially constrained budget environment and with a limited time window to execute its exploration missions. And, along with man-rating requirements, NASA may wish to limit its product scope and variety for these very pragmatic reasons. That might also account for NASA's viewpoints differing from the observations by RAND (below), which also highlighted the front-loaded technology maturation efforts and small procurement numbers of space programs as different from other DoD systems, but still applicable for evolutionary acquisition (Lorell, Lowell, & Younossi, 2006). And in RAND's context, the "space programs" were all satellites—none carrying human life as payload.

Thus, there seems to be an at least perceived aversion to spiral development of (user) life-threatening products such as manned space vehicles (and perhaps pharmaceutical drugs, aircraft, etc.), systems in which long product cycles and much bureaucratic control are often observed as measures to control risk (Dillard, 2005). Aside from truly singular efforts, we have not yet found any universal evidence of the

[19] Feature-length movies can have sequels and pharmaceuticals can have spin-offs, but they are for the most part long product-cycle projects that result from a singular unified approach.

spiral approach being more or less applicable according to quantity of systems produced.

Logistical Support during Service/Shelf Life

Our observations warn that multiple configurations of hardware products do come at a cost for ownership. Veterans of new system deployments across the force/fleet, or throughout any large using organization, know the difficulties of rolling out a configuration change. Benefits of standardization have long been offered via production economies of scale, commonality of parts across platforms, and interoperability. If the ultimate goal is to have standardization across the DoD's force, owning multiple configurations of a system (variety) equates to added complexity in training and supply support of the item. Neither can the logistical maintenance strategy be ignored: whether the end-item is *maintenance-intensive* (such as tactical vehicles) or *maintenance-free*—such as with many electronics items and munitions, and situations in which physical changes are completely transparent to the user. For multiple product configurations, the answer could have a huge effect on the total costs of ownership, as previously mentioned by RAND in regard to UAVs.

Range of Requirement Attainment

Most requirements are "continuous," i.e., may be satisfied in varying amounts of attainment. Thus, ranges of their satisfaction can be flexibly specified, allowing for thresholds (minimum values of attainment) and objectives (optimal values of attainment). Examples are range, accuracy, weight, reliability, etc. However, we have found that some requirements, often critical ones, are more *binary* in nature than *continuous*. They have a much narrower range of attainment, such that they are almost pass/fail or go/no-go in their demonstration. Examples are soft launch, network security, physical fit, leak-proof, shock/vibration/drop proof, survivability, horizontal-to-vertical flight transition, etc. If one of these more *binary*-type requirements happens to be a key performance parameter, its attainment will be on the project's critical path and highly dependent upon technical maturity. As such, it may practically dictate the length of the entire advanced development effort and make division into capability increments less beneficial as a development strategy. Such was the case of Javelin's "soft launch" requirement, described in the case below, where attainment was dependent upon precise timing of ignition and dual-motor burn, facilitated by electronic fusing and solid rocket motor-propulsion technologies. Though strongly correlated with product reliability, these kinds of requirements demand a system that "either works or it doesn't"—without flexibility.

Amount of Change—and the Lure of Modularity

These authors subscribe to the current theorists' view that system complexity is comprised of numbers (of components), connections (interdependencies) and distinctions (variety). Distinction corresponds to variety, to heterogeneity, and to the fact that different parts of complex systems behave differently (Heylighen, 1997). Variety is a component of Nobel Prize winner Herbert Simon's explanation of complexity—many *different* parts with many interactions. Simon argues, from his observation of complexity in things both natural and artificial, that complex systems evolve from simple systems. And they do so more rapidly when there are stable, intermediate forms or sub-systems (like modules or "units of action") (Simon, 1981). Moreover, he argues the resulting evolution into the complex system will be hierarchical. In "*The Architecture of Complexity,*" Simon proposed hierarchy as a universal principle of complex structures. He felt that complex problems could be solved more easily when decomposed into sub-problems (much as how we employ Work Breakdown Structures (WBS) via the Systems Engineering Process (SEP)). And sub-problem solutions could be combined into a solution to larger problems, etc. His famous "parable of the watchmakers" illustrated his hierarchical architecture principles and the benefits of employing modular subassemblies versus elementary components (Simon, 1962).[20] Commonly seen today are modular industrial products that are sometimes designed as complete architectures, with standardized interfaces that invite others to introduce complementary products for insertion (Agre, 2003). The Modular Open Systems Approach (MOSA) is a relatively new DoD initiative that encourages the use of widely supported commercial interface standards and disciplined interface controls to develop systems architectures using modular design concepts (DoD Open Systems Joint Task Force, 2003, August). But despite attempts over the last two decades to “architect” the command, control and computers (C3) domain with initiatives (such as compulsory use of Ada as a high-order software language and imposition of a Joint Technical Architecture (JTA)) as ways of achieving interoperability, a plethora of incompatible “stovepipe” solutions nevertheless continue to proliferate in an almost chaotic evolution (Greene, 2007, March 1). This may be in large part because of the continuing realities of different services or communities with differing concepts of operations (CONOPS) driving different system requirements with different funding streams and different timelines.

As in biological evolution, improved “fitness” with a system’s environment is what is sought in the adapting or evolving of systems. But others have noted that Simon’s metaphors for dynamic complex systems, useful as they are for understanding complexity, fall short of explaining their *evolution*. While the concept of modularity suggests approximately independent subsystems may be modified or adapted as such, it has been shown that, in the aggregate, there is yet quantifiable

---

[20] In his imaginary story, watchmakers named Hora and Tempus were highly skilled watch builders. But Hora prospered more than Tempus because of differences in their watch designs. While each maker’s design was comprised of 1000 elementary components, Tempus's watches weren’t hierarchical, and were assembled one part at a time. But Hora's watches were organized into hierarchical subassemblies of ten parts each. He could combine ten of these subassemblies into larger subassemblies, and then ten of these, until a complete watch was formed. The difference in the two watchmakers’ designs was evidenced when customers interrupted them throughout the day. When this occurred, they would put down their work and their uncompleted watches would fall apart. These interruptions didn’t disturb Hora, who lost at most ten units of work for whatever subassembly he was working on. Tempus, however, would typically lose much more, as he had to start all over with individual parts versus modules. Simon illustrated that that Hora could complete many more watches than Tempus over time, given the usual interruptions that both would likely experience.

modular interdependency that affects evolvability (Watson & Pollack, 2005). In other words, how changes in the state of one module affect the state of another is relative and measurable. Simon's watchmaker parable illustrates that modularity is beneficial for *production*, but not necessarily for *evolution*. Examples of modular interdependency are plentiful. In the aircraft or automotive realm, an engine upgrade would seem intuitively to be a relatively independent subsystem change. But systems engineers know that changes propagate through hardware almost as much as software *in the long run*—just as the eventual rise in building temperature from the thermostat adjustment in one modular room.[21] Adding increased armor protection (and weight) for deployed High Mobility Multi-purpose Wheeled Vehicles has resulted in increased wear-out of drive train and suspension components and impacts to vehicle range, mobility, mileage, etc.—so that "up armor" kits have become only a stopgap measure until totally re-designed systems can be produced. Similarly, the 2006 engine upgrade of the CH-47F helicopter is more of a total system refresh: "95 percent is a new airplane," according to Boeing Defense Systems, despite exterior appearances.

Thus, we suggest it is not only the focus upon structural modularity as such, and standard interfaces, that enable systems evolution. Rather, it is the relative interdependency of the modules. In short, PMs need to be mindful of the degree of change in subsequent increments/spirals. One subsystem is likely to affect another in the short- or long-run. And that can make product evolution problematic. As Norman Augustine once said, "No change is a small change"; independent subsystems, even redundant ones, aren't always independent (Augustine, 1997, June).

[21] Systems theorists have long used the thermostatic example of a cybernetic system feedback loop.

The RAND Study of Evolutionary Acquisition in DoD Space Programs

In our literature review for this research effort, the authors examined the 2006 RAND Corporation report under its Project Air Force series entitled, *Evolutionary Acquisition—Implementation Challenges for Defense Space Programs*, by Mark A. Lorell, Julia F. Lowell and Obaid Younossi. Their research principally addressed DoD space programs and focused primarily on program costs and the cost-estimating implications of evolutionary acquisition strategy. Their methodology consisted of literature review, interviews and five case studies. The program cases were:

- Space-based Space Surveillance (SBSS) System
- Rapid Attack Identification, Detection, and Reporting System (RAIDRS)
- Global Positioning Satellite (GPS) III
- Space-Based Radar (SBR)
- Kinetic Energy Interceptor (KEI)

RAND cautioned that these programs were all in their very earliest stages and that lessons derived from them must be considered tentative. We noted earlier that these were all non-man-rated systems.

In their research, RAND's objectives were similar to ours: seeking to ascertain programmatic implications, lessons already learned in recent space programs, and methods to adopt for effective implementation of evolutionary acquisition. They were careful to distinguish DoD space programs as different from other acquisition programs in at least four important respects:

1. They are characterized by very *small procurement numbers* of end-items (space vehicles)—typically 1 - 25 satellites (with 6 being average), compared to much larger procurement numbers for products such as tactical aircraft or smart munitions.

2. Space vehicle component testing cannot be done in a true operational environment (space) because of the high cost of space launches and the limited number of operational space vehicles in any system.

3. A larger percentage of total program expenditures take place in the early phases of a space acquisition program in contrast to other acquisition programs.

4. Space program technology development extends longer into the procurement process than is typical for other types of programs and has been formalized in the National Security Space Acquisition Policy 03-01 (NSSAP 03-01) regulations. (Lorell, Lowell, & Younossi, 2006)

Their acquisition management findings were:

1. *"The new DoD guidance regarding evolutionary acquisition (DoD 5000 series and NSSAP 03-01) permits great flexibility, but does not eliminate conceptual and definitional ambiguity. As a result, evolutionary acquisition programs vary considerably in their practical implementation approaches"* (Lorell, Lowell, & Younossi, 2006). Program Managers that RAND interviewed perceived having more flexibility to tailor their program structure and technical approach. But confusion and inconsistency still persist among programs they observed (terminology, feedback loops, etc.). Also, most programs are still focusing upon the initial project increment, and often there was pressure for end-state capabilities in the first spiral—causing programs to become more like single-steps-to-full-capability. However, to these authors it comes as no surprise that the advanced capability most needed is likely to depend on the offerings of the least mature technology or binary-type requirements. And we shall later illustrate with a case from our own experience.

2. *"All of the case studies point to the conclusion that the capabilities and requirements definition and management processes are major challenges in all EA programs.* Appropriate structuring of evolutionary acquisition phases with operationally useful threshold requirements and mapping the path to overall objective capability are demanding tasks on most evolutionary acquisition programs" (Lorell, Lowell, & Younossi, 2006)

3. *“The use of the officially preferred spiral development process for implementing evolutionary acquisition on major hardware acquisition programs greatly increases the level of program uncertainties, raising serious challenges for program managers in the current acquisition environment”* (Lorell, Lowell, & Younossi, 2006). The open-endedness and uncertainties of *evolutionary acquisition* that offer valuable flexibility are proving to be politically impractical, especially for large, high-visibility programs. Smaller programs get less scrutiny and could be more flexible, but even they have demands for definitive budgets—within an inflexible PPBE system that is incongruent with spiral policy tenets. The uncertainties of future requirements and technologies greatly challenge the validity of life-cycle costs (LCC) estimates, and with increasing up-front and on-going cost analyst community workload. “Evolutionary costing” appears to be speculative and could give rise to allegations of less-than-full disclosure. RAND also observed that, “There is no question that increased program complexity is an inherent attribute of the evolutionary acquisition approach. This is because evolutionary acquisition envisions multiple increments, all of which are treated in a management sense as quasi-separate programs, with their own milestone reviews, oversight documentation, and so forth. This complexity is increased by the tendency to move (program) content around from one increment to another” (Lorell, Lowell, & Younossi, 2006)

The RAND authors pondered the applicability of evolutionary acquisition to “large-scale hardware” programs, saying the data and analysis is still incomplete on non-space Major Defense Acquisition Programs. They reiterated the differences between DoD space and non-space programs, but extended some of their findings to other programs in general. They summarized the views of non-space program office officials interviewed as:

> A cost-effective program requires *stable requirements, system configuration, and quantities, and adequate funding*. In their view, evolutionary acquisition and spiral development approaches promote constant flux in all these program attributes, leading inevitably to cost estimating difficulties and cost growth. The definition of program content in the Global Hawk (UAV) program, using spiral development and user feedback “created continuous change and uncertainty in all aspects of program management and cost analysis. According to the Global Hawk prime contractor, the program has experienced unprecedented levels of ‘requirements churn’ (Lorell, Lowell, & Younossi, 2006). ”The key lesson learned from Global Hawk, according to one official,

is that the only way to implement spiral development effectively was *to provide unlimited funding to cover the unending changes*" (Lorell, Lowell, & Younossi, 2006; emphasis added)

Thus, RAND highlighted the evolutionary acquisition challenges of requirements and technology churn, spiral or increment definition and content, program complexity and concurrency, logistics planning and density, funding coordination for increments, the regulatory environment, and oversight requirements. These are challenges in any program, but RAND feels (and these authors agree) that evolutionary acquisition presents the opportunity for them to be even more formidable. The RAND study validated several of our previously published concerns about evolutionary acquisition and is predictive of others (i.e., funding challenges and uncertainty, organizational stress, excessive regulation and scrutiny). However, as with most aspects of program management, there are trade-offs to be made and balances that must be struck.

Anecdotal Clues for Coping with Variety and Complexity

With chaotic and uncontrolled change, we envision the risks of unpredictable and disruptive interactions between agents and environments. But all change is not disruptive or negative. We might need only to look at our experience to realize some hints about beneficial variety and successful control of change. One of the most visible examples of product (and capability) variety of late has been in the small arms arena, where a plethora of individual weapon configurations are seen in the many photographs of troops deployed in Iraq and Afghanistan. Soldiers are able to individualize their weapons with infrared aiming devices, flashlights, forward pistol grips, telescopic and illuminated optical sights, etc. (See Figure 8.)

Figure 8. Variety of Individual Combat Weapon Configurations

Such was not the case until the advent of war following the September 11th, 2001, attacks in New York City and Washington DC. Prior to that, configurations of the M16A2 rifle were standardized among Army units, such that the benefits of an optional telescopic sight and mount were considered too burdensome for logistical and combat command and control at troop level. However, from dozens of informal interviews of returning officers, the collective explanation of how deployed units are able to manage variety in the field is via individual *ownership and accountability*: The troops are now issued a rifle in basic training that accompanies them throughout their entire combat tours. They are strictly *accountable*, more than at any time in the recent past, for their own configuration and operator maintenance of their weapons. (see Figure 9.)

Figure 9. American Soldiers are Accountable for their Individual Weapons upon Entry (*Army Times*, 2007, February 12)

In the same way, much higher levels of risk from system complexity are generally believed to be mitigated by control measures, as within organizational contingency theory (i.e., centralization/decentralization, etc.).[22] The American nuclear Navy was rooted in Captain Hyman G. Rickover's visit to Oak Ridge National Laboratory in 1946 to investigate the feasibility of using nuclear power aboard submarines. During his long tenure as head of the nuclear program, he maintained fundamental principles about technical and organizational program structures, not

[22] The theory holds that organizational structures must change in response to contingencies of size, technology, and as external environments become more complex and dynamic. See Woodward, J. (1958). *Management and technology.* London: HMSO.

the least of which was personal *accountability*. During his testimony before Congress about a nuclear accident at the Army's Stationary Low-power Plant Number 1 in Idaho, which killed three technicians, he said:

> I have many people carrying out tasks in the program and I hold them *accountable* to me for those tasks. But if anything important goes wrong in my program, is there any doubt in your minds who is responsible? I will tell you right now, in case there is any uncertainty about it, I am responsible. (Rockwell, 1992)

Those who have worked with acquisition of nuclear plant materials know well both the specifications and standards of quality unique to this commodity as well as Rickover's tenets of responsibility and accountability that still exist today. It is largely believed to be one important aspect of how he successfully dealt with the complexities and uncertainties of a new application of technology.

Another recent example of successful control of rapid change lies in the Acoustic Rapid Cots Insertion/Advanced Processing Build (A-RCI/PB). In this vital program for sustainment of submarine acoustic sensing superiority, a series of hardware and software upgrades were planned and executed in rapid succession. Each emerged with advancement in capability, keeping pace with technology and competitive threats, facilitated by *rigorous control of interfaces, standards and protocols* (Boudreau, 2006).

Observations and Realizations from Historical Cases

History reveals that spiral development for large complex hardware systems can be a successful approach. One of these authors was fortunate to have helped lead the development effort on two fielded missile systems that are now combat-proven and still in production: the Army Tactical Missile System (ATACMS) that premiered in the first Gulf War and the Javelin anti-tank missile now being used in Iraq and Afghanistan. Both were born out of DARPA initiatives and became major acquisition category (ACAT) 1D (OSD-level review) programs. And both have experienced variety and change, but with very different acquisition strategies.

ATACMS—Incremental and Spiral Development

Figure 10. The Army Tactical Missile System Components Launcher, Missile, and Missile Launch Pod Container

The Army Tactical Missile System program successfully used evolutionary acquisition with both incremental *and* spiral approaches. In the 1980s, the Army sought to achieve an organic deep-strike (greater than 100km range) capability by expanding the use of its Multiple Launch Rocket System (MLRS) platform (about

40km range) with a semi-ballistic missile. (See Figure 10.) An anti-personnel/materiel missile would be developed with each one containing roughly 1000 one-pound bomblets. The weapon would, in the next increment, be further enhanced by evolving to a warhead that could dispense "smart," or guided, submunitions. This was viewed as a simply articulated, pre-planned product improvement (P3I) acquisition strategy that incrementally attained fully envisioned requirements, separated into blocks of capability. (See Figure 11.)

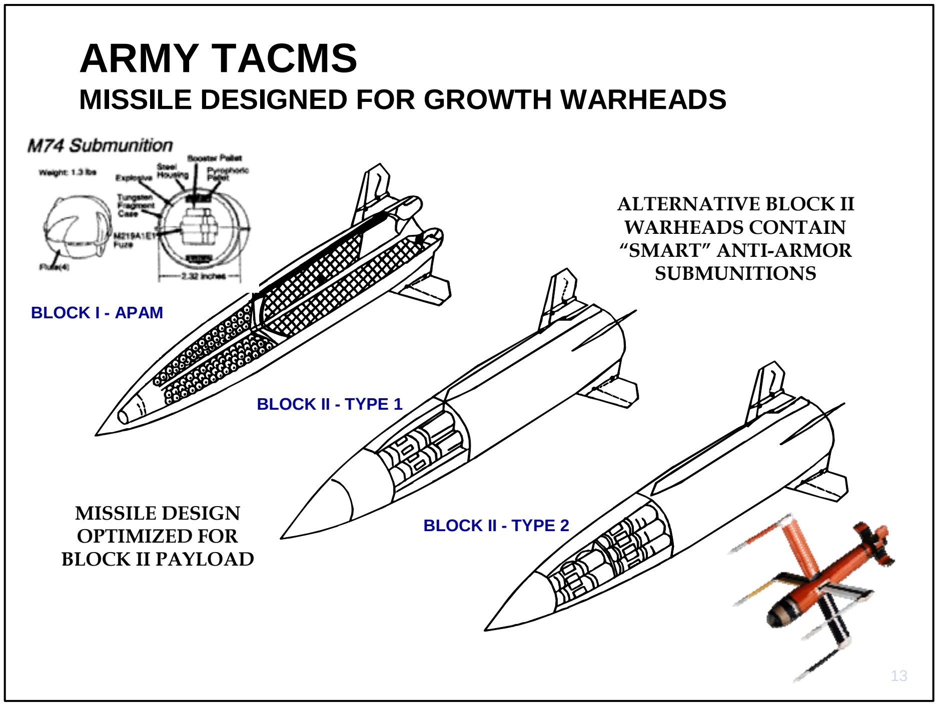


Figure 11. Army TACMS Program Strategy Visual Depiction

The program office commenced a 48-month advanced development effort in 1986, skipping a technology development phase, and using a pair of Firm-Fixed Price (FFP) contracts: one for invention of the missile and one for its integration into the MLRS platform (with the same contractor—LTV Corporation—prime vendor of the platform). Critical technologies in the initial capability Block I (M39) were: solid rocket motor propulsion, fusing of bomblet dispense and detonation, missile and launcher navigation, software for missile guidance and launcher fire control. All were assessed to be mature (although today's Technology Readiness Level rubric did not yet exist). A Honeywell navigational laser ring gyroscope that was employed in Boeing 727 aircraft was used for the missile guidance set. M-74 bomblets and fuses from decommissioned LANCE conventional missiles were downloaded and used as government-furnished material (GFM) for the warhead. Mechanical safe-arm fuses were dually used for warhead dispense and warhead severance packages (later evolving to electronic safe-arm fuses). Missile hardware component size and weight were only constrained by the limits of the MLRS platform's architecture, and a requirement for handling and external appearance similarity with the shorter-ranged rockets it was replacing. Launcher modifications included additions and modifications to several line-replaceable unit (LRU) components—again, most fitting easily and as relatively independent modular components within the platform architecture. They augmented electronic power and its distribution to the launcher system and improved launcher position determination.

Mature Technology Shortens Product Cycle-time

ATACMS entered low-rate initial production a full year prior to operational testing and evaluation, based upon accomplishments during development testing. The Block I program finished on budget and culminated in a highly successful operational test, still using development units as test articles, and extending only three months beyond the 48-month contract period. Four months later, the full-rate production pricing options were preserved when ATACMS was approved for full-rate production by OSD-level review, only one week prior to their expiration. ATACMS entered the Persian Gulf War with its operational test unit firing about 32 production missiles in combat (Redstone Arsenal, 2007).

Truly, this was a low-risk program that was structured commensurately. One of its key lessons was that even though it was an entirely new product, the extensive use of mature technology eliminated at least one development phase, greatly shortening cycle-time to deployment and enabling the use of a contract type normally reserved for the certainties of production. It was never envisioned to grow into an evolutionary family of many different missiles[23] from planned and unplanned developmental increments (more than 450 of which have been fired during Operation Iraqi Freedom). There were other lessons to be learned from this program as well. In terms of ex poste facto “product discovery,” the Joint Force Air Component Commander for the Korean theater in 1995 surfaced an issue of service “ownership” of the recently deployed ATACMS capability. Despite its years in development (and with initial US Air Force participation in its requirements generation and program formulation), ATACMS’s ability to engage target sets that were previously only within range of USAF aircraft was not yet fully realized by all components. This led to a revisiting of service roles and missions within the theater. From a product-development perspective, *an elegant and open architecture* enabled a series of planned and unplanned system variants to emerge.

Planned and Unplanned Variants

A low-level, internal technology development program had been conducted by the same program office in parallel with the ATACMS development project. It used a subordinate product manager and matrix personnel from within the PMO and supporting R&D community. It was an real option to fulfill the vision of a Block II anti-armor capability. However, what actually became the smart submunition for ATACMS, thirteen of them in each missile, was the Brilliant Antiarmor Submunition, or BAT. The ATACMS Block II (M39E3) BAT (originally for Brilliant Anti-Tank) smart submunition program was quite a different program and employed a different contractor (Northrop Grumman). After a lengthy technology development effort (1984-1991) under a separate program office, BAT entered advanced development as ATACMS went into full-rate production, and later merged with the ATACMS program office (in 1994). The BAT was to employ both acoustic and infrared (IR)

[23] There was, however, a vision of an MFOM (MLRS Family of Munitions)—both rockets and missiles—to be fired from the versatile carrier, but not so many variants of the one ATACMS missile.

guidance and, upon release from the ATACMS carrier, to glide aerodynamically and autonomously attack mobile armored targets (GAO, 1997, October). Among the critical technologies for its capability were acoustic sensor, infrared seeker, tandem shaped-charge warhead, and digital processing. It was to enter low-rate production in 1995 after 40 months of development effort. It finally did so in 1998 after significant cost and schedule overruns (GAO, 1999, July 30, p. 5). Highlighted in the GAO's report on DoD's pursuit of immature technologies during advanced development, these were cited as "main contributor(s) to the program's 88-percent cost growth and 62-percent slip in schedule." The BAT program, while an example of incremental pre-planned capability growth and parallel development, serves perhaps as a better example of over-ambitious scheduling and flawed cost estimation. Nevertheless, the capability of deep-attack anti-armor was eventually added to the Army's portfolio of needed capabilities, and the submunition itself was also incrementally improved via P3I.[24]

Spiral development came into play for the ATACMS with the proliferation of Global Positioning Satellite (GPS) technology, and when post-Persian Gulf War analysis revealed a need for an even longer-range strike capability. These feedbacks from the technological environment and user community drove an innovative development approach to attain a substantial extension in ATACMS range and with precision accuracy. GPS augmentation of the standard missile guidance set reduced circular error probable (increased accuracy), and allowed for a reduction in bomblet payload (by over 600 bomblets) such that the range could be extended to well over 250km. These "unplanned" system improvements took place while the Block I system was in full-rate production, and Block II was still under development. Block IA (M39A1) entered low-rate production in 1996 and 1997, with full-rate production in 1998. Though not touted as such until now, this initially

[24] A BAT P3I (M39E4) program, funded through 2002, provided a new sensor suite with millimeter wave and imaging infra-red to the basic BATs acoustic and infra-red sensors. It improved inclement-weather capability and effectiveness against countermeasures, along with expansion of the submunition's target set.

undefined and incremental change in system configuration and operational capability epitomizes the philosophy of hardware spiral development in acquisition.

Again in December 2000, and as a result of Kosovo lessons learned by the Army in 1999, a Quick Reaction Program was initiated to rapidly attain another tactical capability—that of a large unitary munition. Designated the Army TACMS Block IA Unitary missile, a development contract was awarded to Lockheed Martin (formerly LTV until 1992) to employ another GFM munition—this time a proven unitary warhead from the Stand-off Land Attack Missile (HARPOON WAU-23/B)—to be integrated into the Army TACMS Block IA missile. The first missile was delivered within four months after contract award, with 41 more produced through the end of 2001. Program supporters said the rapid achievement, "clearly demonstrate(d) the versatility and agility of the Army TACMS design" (Lockheed Martin, 2001, April 23).

Changes in technology and user needs gave birth to yet another ATACMS variant in the 2001-2005 timeframe: the ATACMS-P, or Penetrator. This is a standoff ballistic missile, delivering an earth-penetrating warhead for use against fixed hard and deeply buried strategic and tactical targets (US Army RDT&E, 2004, February). This system is employed from both the M270A1 MLRS platform and the newer, wheeled vehicular High Mobility Artillery Rocket System (HIMARS). The ATACMS-P began as a Joint service Advanced Concept Technology Demonstration integrating the Army TACMS booster with a Navy Strategic Systems Program (SSP) re-entry vehicle built by Sandia National Laboratories. Funded under the BAT P3I RDT&E line, it was conceived for attacking high-value targets that were perceived threats to US and coalition forces in the post-9/11 campaigns. Successful test flights in March of 2004 and August of 2005 demonstrated test objectives of booster separation and ballistic flight path of the penetrator to its target.

An Architecture for Variety, and the Need for Control

In all, these ATACMS program variants comprise a validated chronicle of operationalized evolutionary acquisition over more than two decades. While not applicable to all programs, perhaps because of each systems' unique product attributes, these multiple product releases show at least the ability to respond to changing technology and user needs given time, funds, and a simple architecture that can accommodate change. Similarly, other large ground vehicles, naval vessels and airframes in particular, because perhaps of their larger frames, seem to accommodate modular upgrades easily. As alluded to earlier, some munitions also lend themselves somewhat to variety without some of the usual attendant support costs because of their "wooden" nature—a term used to describe maintenance-free end-items. "Deep Attack" modified MLRS launchers did indeed have relatively independent modules and open critical interfaces, for electrical power supplies, navigation, fire control subsystems, etc. For optimal emphasis and control, the vehicle integration effort was considered to be significant—thus, the separate contract for it.

Interestingly, variety proved itself a menace to the ATACMS program after production was initiated. A change in the ATACMS Block I production design resulted in a rocket motor nozzle burn-through, discovered during production-verification testing. Failure analysis concluded that a material specification was insufficient for the application, but wasn't evidenced until a change of component suppliers. Moreover, the failure revealed both government and contractor had *insufficient configuration control* when uncertainty arose over which missiles had the deficient component. This small change—to save only $15.00 per missile—necessitated a very expensive retrofit of dozens of missiles (Army TACMS Project Office, 1993, May 14) (Figure 12).

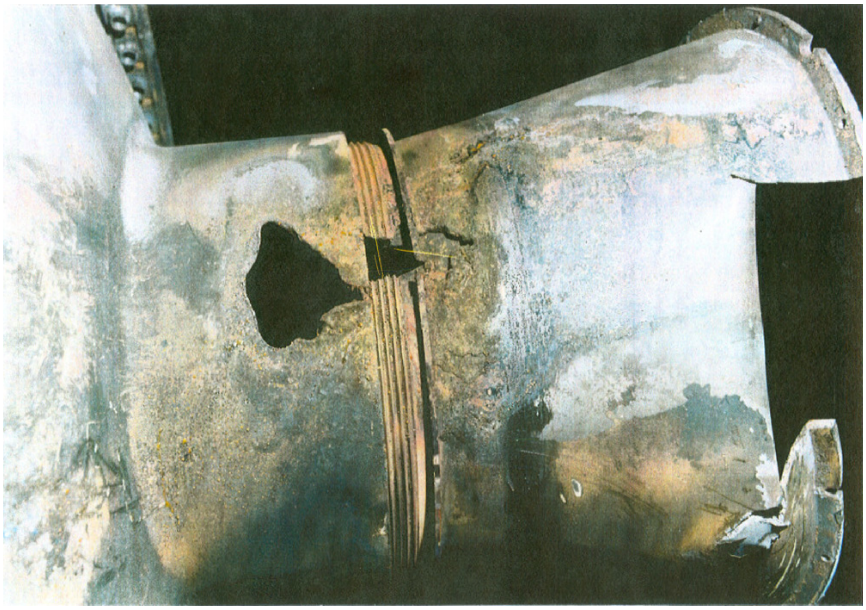


Figure 12. ATACMS Nozzle Exit Cone Assembly Burn-through

While the only officially recorded test failure from this cause was at the White Sands, New Mexico Missile Range, anecdotal evidence from a returned Persian Gulf War explosive ordnance disposal specialist indicated at least one of the munitions fired there experienced the same failure mode, and is thus believed to have been from the deficient lot (Matthews, 2006, December).

Figure 13. The Javelin Anti-tank Weapon System (Missile and Command Launch Unit)

The Advanced Anti-Armor Weapon System—Medium (AAWS-M), later to become the Javelin, began in 1982 as the DARPA program “Tank Breaker” (stinet.dtic.mil) (See Figure 13.) This was a one-year technology demonstration to explore various missile guidance solutions for a medium range (i.e., 1-2000 meters), man-portable, anti-tank weapon. It was spawned as a result of deficiencies that were immediately apparent in the newly fielded DRAGON weapon system, which had replaced the M67 90mm recoilless rifle in the late 1970s. The DRAGON was a wireguided line-of-sight missile that was developed in response to the 1960s appearance of the Soviet AT-3 SAGGER, a manpack missile carried in a fiberglass "suitcase." In 1978, a Mission Need Statement highlighted deficiencies of the Dragon, such as its poor reliability, limited range/lethality, and the difficulty for gunners to aim and track targets. The envisioned replacement was to satisfy a substantial increase in requirements, namely: range, lethality, reduced weight, and the ability to launch from enclosures (such as buildings or field-fortified bunkers). Several years were spent finalizing these requirements until the joint Army and Marine Corps operational requirements document was formally approved in 1986-88. A competitive fly-off program, which would now be called the “Technology Development phase,” was conducted in 1987-1989 to select from three teams of contractors and critical technologies: a laser-beam rider led by Ford Aerospace, a fiber-optic guidance effort led by Hughes, and a forward-looking infra-red (FLIR) thermal imaging sensor effort from Texas Instruments and Martin-Marietta. Cost-plus-fixed-fee (CPFF) contracts were used with each of the three teams. All three teams were successful in flying missiles to their targets, but the only technology that enabled a true fire-and-forget capability (which was not a specified requirement) was the Forward-Looking Infra-Red (FLIR) approach, enabled by a comparatively new technology: focal plane arrays (FPA). Though this approach was recognized to be the most technically immature and risky, the desire for fire-and-forget survivability resulted in the FLIR team being awarded a contract for a three-year advanced development phase.

In June of 1989, a full-scale development (now called System Development and Demonstration) contract was awarded for the AAWS-M project. At the macro level, the office of the Secretary of Defense viewed the program as acceptable with regard to risk because of its 27-month technology development phase, use of real options for a technical solution, and subsequent 36-month plan for full-scale development. At the program office level, it was known to be one of high risk in several technical areas.

Immature Technology Lengthens Product Cycle-time

Technology risks were adjudged to be in the following areas: focal plane array producibility (from the standpoint of specified temperature sensitivity), tandem warhead performance (pushing the physical limits of armor penetration versus package size), software tracker algorithm (to maintain a target lock with optical correlation of target characteristics supplied by the FLIR), two-stage rocket motor (which would enable “soft launch” from enclosures), electronic fusing (timing in micro-seconds for the dual warheads and dual motors) and system weight (also pushing the physical limits of cubic dimension) (Lyons, Long, & Chait, 2006, July). All of these technical risk areas would be considered as immature by today’s TRL standards (see Figure 14).

During the technology development phase, all three contractor teams had scored over 62% hits with at least ten missile shots each in a variety of environments and operational settings. The full-scale development contract request for proposal was written for a cost-plus-incentive-fee type of contract, giving incentives for key performance parameters such as weight and warhead performance considered to be technically risky. The total value of the contract was $169.7 million, the amount bid by the winning team of Texas Instruments and Martin-Marietta, who formed a Joint Venture. Meanwhile, the Government privately conducted its own should-cost estimate and budgeted $263 million for the thirty-six month long advanced development effort. In addition, the Government ran its own alternate warhead technology development program with Conventional Munitions Systems (CMS) acting as the contractor.

The two-partner Joint Venture in full-scale development was also free to maximize competition at the subcontractor level. In their make-versus-buy decision, Texas Instruments elected to make the focal plane array for both of its uses in the command launch unit and in the missile. The company had made these devices for other programs, but not in these two distinct configurations. Focal plane array technology was still immature and would be gauged today at approximately technology readiness level 5 (on a 1-9 scale) despite its successful technology-development phase results. It was always recognized as technologically risky, so the Government funded its own night-vision laboratory to partially fund other companies that could produce these devices. In 1991, the only five known FPA makers in the world were: Rockwell International, Loral, Santa Barbara Research Corporation, Sofradir (a French firm), and Texas Instruments.

As an additional gauge of technological maturity, a comparative baseline test was mandated at the second milestone upon the decision to launch the Javelin program into full-scale development. That test would pit the immature focal plane array technology against existing TOW and Dragon (legacy systems) night-viewing optics. Results of this test showed the Javelin's immature focal plane arrays to be substantially better in performance than the Dragon and almost as effective as the larger TOW anti-tank missile system.

However, approximately eighteen months after the full-scale development-phase contract award, the Javelin project manager forecasted a Nunn-McCurdy breach of cost and scheduling thresholds in this ACAT 1-D program, and called for a non-milestone Defense Acquisition Board review. Several reasons were cited; chief among them was that the focal plane array production yield was not as predicted, and all of the devices were below specification. Weight was also a significant contributor (even after a Joint Requirements Oversight Council (JROC) approved requirement threshold change (from 45 to 49.5 pounds)), causing redesign of many components for reduction.

Over the next year, the program sought a new baseline with many different revised program estimates—climbing from 36 months duration and $298 million in cost, to 48 months duration and $372 million in cost, and finally 54 months and $420 million for the total cost and duration of this phase. Within that year, the program was restructured, given the new baseline, and finished largely within its new parameters. The additional eighteen months added to the 36-month phase helped resolve the uncertainties and complexities of system development without additional schedule slippage. Later, production quantities were slashed in half as the Defense Department drew down its forces from 1991-2000, and the acquisition strategy to split apart the Joint Venture and compete them in production was not fulfilled. Benefits of a split production no longer able to be realized, the Joint Venture remained intact as the producing entity.

Unplanned Variety and the Need for Control

The GAO was harshly critical of the Army's plan to enter a multi-year contract (seeking to stabilize contractor workload and achieve economies of scale). After several years of Low-rate Initial Production (1994-96), the GAO stated that, "The Army has not demonstrated that Javelin's *design is sufficiently stable* for a multiyear contract" (GAO, 1996, September; emphasis added). But the Army proceeded to enter multi-year contracts in 1997 and 2000, despite at least 30% of all system components experiencing redesign during low-rate production.

Moving to performance specifications under the last acquisition reform era (1994-99), the program began to relinquish configuration control to the contractor and saw continuing redesigns for virtually all system-configuration items. Like the GAO, the program management office also sought design *stability* and had significant concerns over a continually changing production baseline. The program management office realized during this period that the Government must be accountable for prescribing the entire system's performance margins and remain vigilant to insure the contractor doesn't "trade off" hard-won design margins to lower unit costs (Knox, 1999, September). This was found to be especially true in technical areas that can seriously impact operations and support cost/performance. Similarly, it is not always possible to realistically test the contractor's compliance with performance requirements and whether the system is still operationally effective and suitable. Communication and *trust*, with verification, are necessary facets of the government-industry partnership (Zolin & Dillard, 2005, May). And some entity still must own, maintain, and be *accountable* for a technical data package for the entire system.

Acquisition reforms were not intended to remove discipline, but to eliminate non-value added bureaucracy. As with the ATACMS rocket motor case failure, there must be strict configuration control. And as practitioners are expected to know, configuration management is not for the prevention of change, but rather for *controlled*, approved, and documented change. Used appropriately, it provides a disciplined approach for managing change to a system's design so that any change is analyzed—from a system and life-cycle perspective—for its potential impacts.

The Javelin program had always planned to employ interim contractor logistics support enroute to some eventual level of organic system support (principally of its target acquisition device, not of the munition). Since the Javelin's design was in such a state of flux, and an organic stockage of spares therefore impractical, the best approach may have been to purchase spares from a contractor-generated representative spares list and allow for just-in-time delivery. Though the government in fact bought more support than needed, this idea is commensurate with the contractor's control of the configuration and its susceptibility to change without government approval (or even knowledge). Today, Javelin is viewed as being a totally successful weapon system despite its earlier programmatic shortcomings. It is being used in combat operations and has been through several full-rate production contract periods. Over 1000 Javelin missiles have been fired in the Iraq War and Afghanistan since March of 2003, with close to 98% reliability. The system design has continued to be upgraded—not as blocks of capability—but with software, warhead and producibility enhancements; the design of the Javelin has become very "evolutionary" indeed—but not in the manner of evolutionary acquisition's "planned increments of capability."[25]

[25] Acknowledging however, production variants FGM-148A, B, and C; see DoD 4120.15-L, "Model Designation of Military Aerospace Vehicles," 05/12/2004.

Synthesis of the Cases

Our concise cases here only demonstrate that leap-ahead capabilities can result from different acquisition approaches. But it would be difficult to assert that a spiral development approach could have been taken with Javelin that would have resulted in the same capability leaps, or even earlier delivery of some lesser capability, since many of Javelin's key performance parameters depended upon immature technologies (or binary ones, such as soft-launch), and man-rating. The comparison below provides a summary of key program characteristics in the two munition programs (Figure 14).

Key Program Characteristics - First Increment of Capability

Program Aspects	ATACMS	JAVELIN
DARPA Predecessor	Assault Breaker 1977-82	Tank Breaker 1981-82
Ultimate Capability	*"Deep Attack"*	*"Fire & Forget"*
Critical Technologies & Readiness Levels:		
Munition	9 - Lance M74 Bomblet	5 - Tandem Shaped Charges
Propulsion	9 - Solid Rocket Motor	5 - Two-Stage Solid Rocket Motor
Flight Control	9 - Fin surfaces	6 - Fins + Thrust Vector Control Vanes
Guidance and Control	9 - Inertial	4 - Tracker Software Algorithm
Safe/Arm Fusing	7 - Mechanical	4 - Electronic
Software Function (Target Acquisition, Fire Control, etc.)	6 - Various	6 - Various
Sensor	N/A	5 – Focal Plane Array
Capability Leap Area	Range	Range, Lethality, Survivability
Cost of development	~$700M	~$700M
Contract Type	Fixed Price	Cost Reimbursable
Tech Development Phase	0 Months	27 Months
Advanced Development Phase - Planned	48 Months	36 Months
Advanced Development Phase - Actual	51 Months	54 Months
Total Time in Development	51 Months	81 Months
Advanced Development Phase Contract Cost Growth	0%	>200%

Figure 14. Comparison of Programs Using Different Development Approaches and Technology Readiness Levels

Both programs achieved capability leaps and have performed splendidly in combat operations. Being only two cases, they cannot alone prove our assertions. But they do illustrate that two munition programs of the same acquisition category and timeframe, with very different technology readiness levels and project scope, had two very different project outcomes with regard to cost and schedule. Further, as different as these programs were in product size and mission capability, they help to convey what program managers must realize about spiral development:

a. That it is an approach primarily for reduction of product cycle-time;

b. It is enabled by the advanced development of only mature technologies;

c. That a system's physical properties (mutability), along with other factors such as time criticality and user risk, binary vs. continuous requirements, required maintenance, and modular interdependence, etc., will influence spiral development applicability;

d. That key capabilities may in fact depend upon the least mature technologies;

e. That an "open," or at least elegant, system architecture enables a basis for independent modular variety;

f. And that thorough design specification and configuration-management accountability is essential for managing the complexity of multiple product releases.

There are many other currently deployed systems that have undergone a long series of upgrades. At Appendix A and B respectively are thirty variants (spanning 30 years) of the UH-60 Blackhawk helicopter and ten variants (spanning 50 years) of the C-130 Hercules aircraft programs, shown as a chronology of their product variation and key capabilities added. Of course, these "spirals" have been realized as product upgrades, but they have indeed been the result of user feedback and mature technology insertion. It becomes apparent that all spirals will eventually become defined increments— "mini-programs." They are often then popularly termed as "spirals," despite their definition. But in years past, they have often been implemented as sequential, separate, and successive product upgrades (also as program examples are the CH-47 helicopter[26] and B-52 bomber). But current policy

[26] Chinook helicopters have been product-improved as the CH-47F model. Six were deployed last year with more powerful engines and avionics improvements. The airframe design is more than 20 years old, and the new models have another 20 years of projected service life.

expresses spirals as more concurrent, frequent and continuous. And that may bring about some of the organizational risks we have already, and will further, discuss.

Modeling Evolutionary Acquisition

In this section, we present our work with the simulation of various project scenarios under evolutionary acquisition (incremental and spiral) and a single-step development approach. This modeling further tests the concepts described and discussed above and provides different insights into the impacts of spiral development on acquisition project performance.

The Modeling Approach

A computational experimentation approach to investigating acquisition projects is applied. This approach integrates theory and practice in a computational tool that allows controlled experimentation through simulation. The current work reflects project theory (e.g., the theory of constraints and work flows), product development theory (e.g., rework impacts and work dependencies), and management theory (e.g., resource allocation and information theory). Practice is reflected in the model through the use of case studies as described in the literature cited to build and validate the model structures and the calibration and testing using the acquisition projects described above. A computational experimentation approach provides many advantages over purely laboratory or field-based methods and benefits from several of the strengths of both laboratory and field research. Nissen and Buettner (2004) describe and discuss the computational experimentation approach, and Dillard and Nissen (2007) describe its application to investigating acquisition projects.

The system dynamics methodology was applied for model development and use. System dynamics uses a computational experimentation approach to understanding and improving dynamically complex systems. The system dynamics perspective focuses on the roles of accumulations and flows, feedback, and nonlinear relationships in managerial control. The methodology’s ability to model many diverse system components (e.g., work, people, money), processes (e.g., design, technology development, quality assurance), and managerial decision-making and actions (e.g., forecasting, resource allocation) makes it useful for investigating acquisition projects. Forrester (1961) develops the methodology's philosophy and Sterman (2000) specifies the modeling process with examples and describes numerous applications. System dynamics has been applied to projects for several decades and has built a collection of validated development project structures (Lyneis & Ford, 2007). When applied to projects, system dynamics focuses on how performance evolves in response to interactions among development strategy (e.g., spiral development vs. traditional), managerial decision-making (e.g., scope developed in specific blocks) and development processes (e.g., concurrence). System dynamics is considered appropriate for modeling acquisition projects because of its ability to explicitly model these and other critical aspects of development projects (Ford & Sterman, 1998; Cooper, 1993a;b;c; Cooper & Mullen, 1993; Cooper, 1994). System dynamics has been successfully applied to a variety of project management issues, including failures in project fast-track implementation (Ford & Sterman, 2003b), poor schedule performance (Abdel-Hamid, 1988), and the impacts of changes (Rodrigues & Williams, 1997; Cooper, 1980) and concealing rework requirements (Ford & Sterman, 2003a) on project performance. See Lyneis and Ford (2007) for a review of the application of system dynamics to projects.

The model is based on previously developed system dynamics models of product development in several industries and the military that have been developed and tested over several decades, as described and referenced below. Therefore, the model is founded on well-established and tested components. These previous models have developed structures for many components and aspects of acquisition. However, previous models have not been used to investigate acquisition approaches such as spiral or incremental development as used by the DoD. The current model uses previous model parts to build a project model that can reflect the important features and characteristics of different acquisition approaches. The model is purposefully simple relative to actual practice to expose the relationships between acquisition approaches and acquisition project performance. For example, total resource quantities and productivities are assumed fixed. Simulated performances using different acquisition approaches are, therefore, considered relative and useful for gaining insight and developing acquisition strategies, but not sufficient for the management of specific acquisition programs or projects. This research approach allows the investigation to focus on how acquisition approaches impact project performance.

A Conceptual Model of Incremental Development

The model structure reflects the structure of acquisition projects. The conceptual (high-level) model structure will be described, followed by a more detailed description of how critical acquisition project features are modeled in the formal (computer-simulation) form of the model.

In the model, four types of work flow through each block of an acquisition project: requirements, technologies, product component designs, and products. Within a development block, each type of work flows through a development phase that completes a critical aspect of the project: 1) develop requirements, 2) develop technologies, 3) design product components (advanced development), and 4) manufacture products. The exception is requirements, which also measures progress through the final phase, 5) user product testing. Development phases and information flows in a single block as depicted in the model are shown in Figure 15. Arrows between phases indicate primary information flows. The start of all phases except the development of requirements is constrained by the completion of previous ("upstream") phases. The completion of some requirements allows the start of technology development, reflecting the concurrent nature of this portion of acquisition. Both requirements development and technology development must be completed for Advanced Development to begin. In turn, the completion of Advanced Development allows manufacturing to start. When some product has been manufactured, they are shipped to users for readiness (operational) testing. Figure

15 also identifies the five major reviews within a single acquisition block (A, B, Design Readiness Review, C, and Full Rate Production) at their approximate times during a project. As described previously, these reviews add work beyond that needed to complete the basic products of each phase (requirements, technologies, designs, products, and readiness for use confirmation).

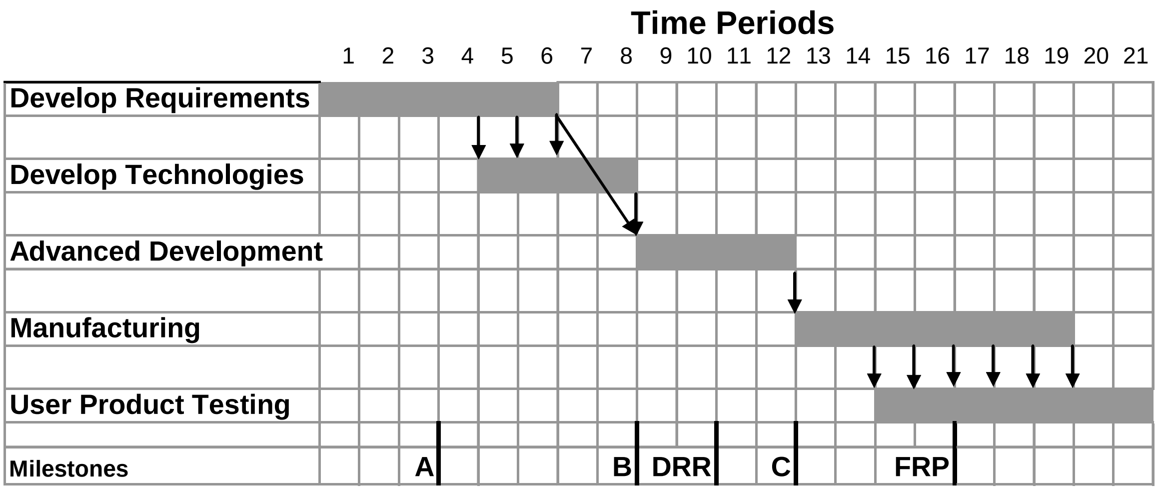


Figure 15. Information Flows in a Single-block Acquisition Project

All development processes are constrained by the physical and information relationships among the activities and phases within a development block. These constraints include development activity durations and precedence relationships, information dependencies leading to iteration (Smith & Eppinger, 1997b), the availability of work (Ford & Sterman, 1998), coordination mechanisms (Hauptman & Hirji, 1996), the characteristics of information transferred among development phases (Krishnan, 1996), and the number, skill and experience of project staff (Abdel-Hamid, 1988). These processes and policies can interact to constrain progress. Even when resources are ample, progress can be constrained by the interdependencies among phases and work packages.

As an example, a development activity that is significantly simpler than most acquisition projects will first be used to illustrate process constraints. Consider the erection of the structural steel skeleton for a single story of a ten-story building. Each member (the columns, beams and bracing) must be installed, inspected, and corrected if the installation is found to be defective. These activities can only occur in a specific order: install, inspect, approve or discover a problem, rework, and re-inspect. When no further problems are found the work is approved and released so other work dependent on that task can proceed (e.g., installation of floors, walls, etc.). In addition to the process constraint imposed by the sequence of activities, if an error is found, the affected supervisors and skilled trades must work together to communicate the problem and devise a plan to remedy it (coordination) before the error can be corrected (rework). Similar processes are used to develop products that are much more complex and unique. For example, the design of focal plane arrays for the Javelin project required an initial design of each component, the testing of the designs (perhaps by review by another designer), the approval of designs for release (e.g., to develop a prototype) or identification of a required change, and retesting. The basic development processes are similar in both the steel beam and focal plane examples. Important characteristics (described next and later) are used to describe important differences.

Development activity durations also constrain progress. For any given technology, a certain minimum amount of time is required for each activity—even when resources are ample. These constraints are captured in the model with specific development activities and backlogs of work in individual phases of an acquisition block (more detail later).

In addition, performing many types of development work depend on the development of other "upstream" work. This availability of work based on the completion of previous work is an important form of progress constraint. Critical path theory models these constraints with precedence relationships that constrain the beginning and end of activities. However, in practice, upstream development can constrain downstream activities throughout their overlapping time, not just in activity beginnings and endings. Returning to the steel erection example, the steel members for the upper floors cannot be installed until the beams and girders for lower floors are in place because the lower floors must support those above. Slow development (installation) of lower floors will constrain the development of upper floors. In the Javelin project, the targeting component design was dependent on the development of focal plane array technology. This type of dependency is captured in our model by precedence or concurrence relationships.

Precedence relationships can constrain progress within (internal) a single development phase or between (external) phases. The feedback structure for precedence relationships within a phase is shown in Figure 16 with a causal loop diagram. In causal loop diagrams, the variable at the tail end of a causal arrow influences the variable at the arrowhead end of the arrow. The polarity at the arrowhead indicates the direction of influence. Positive causal relationships cause the driven variable to move in the same direction as the change in the driving variable. For example, an increase in the Basework (or Initial Completion) rate increases the number of Tasks Completed (ceteris paribus, i.e., all other things held constant or equal), and a decrease in the Basework rate decreases the number of Tasks Completed compared to the number of Tasks Completed if the Basework had not decreased. In contrast, negative causal relationships cause the driven variable to move in the opposite direction as the change in the driving variable. For example, an increase in the Minimum Basework Duration (e.g., minimum time to design a component) would cause a decrease in the Basework rate and vice versa. See Sterman (2000) for more description and examples of causal loop diagramming for modeling causal systems driven by feedback. Causal loop diagrams also identify and label feedback loops. Reinforcing loops (labeled “R”) generate behavior that moves values farther and farther from their initial values in one direction faster and faster. In contrast, balancing loops (labeled “B”) generate goal-seeking behavior.

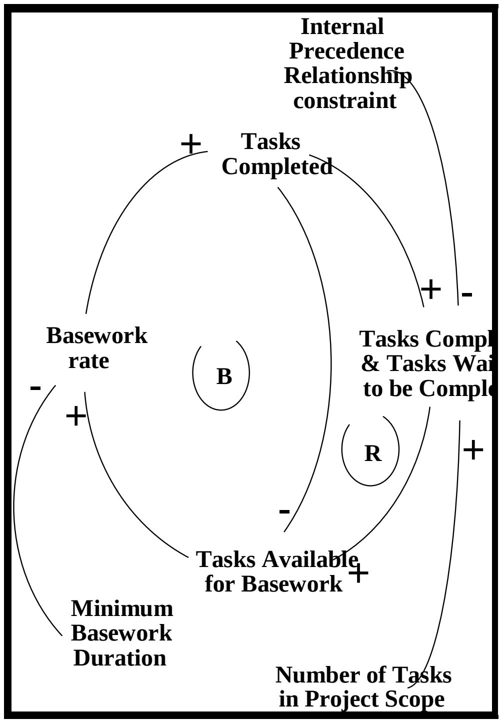


Figure 16. Development Progress Constrained by an Internal (within a Phase) Precedence Relationship

The feedback structure shown in the Figure 16 models the increase in the number of tasks which are available to the Basework activity due to the completion of work. In this loop, an increased Basework rate raises the number of Tasks Completed, which raises the total number of tasks which can be completed. The total number of tasks which can be completed includes both tasks which have been completed and tasks which are available and waiting to be completed. This quantity of tasks is also dependent on the nature of the development process as described by the process's Internal Precedence Relationship. Increasing the number of Tasks Completed & Waiting to be Completed raises the Tasks Available for Basework and, thereby, further raises the Basework rate.

In addition, the Basework of most phases cannot be done without information, materials, and components provided by other upstream phases. For example, the development of technologies depends on requirements information. We capture these constraints through concurrence relationships. Concurrence relationships answer the question, “How much work can we now complete given the work released by the phases upon which we depend?” Reconsider the erection of the steel skeleton of an office building as an example. Erection depends on the release of construction drawings by the design phase and the progress of foundation work (among others). They would be captured in the model by external (inter-phase) concurrence relationships: one describing how much of the steel can be erected based on the release of construction drawings and another describing how much steel erection can proceed based on the state of the foundations. Either of these relationships might constrain steel erection: steel for the ground floor cannot be placed until both the foundation is complete and construction drawings for the ground floor are released. Each external concurrence relationship describes the fraction of a phase’s total scope that can be completed based on the fraction of work released by a supplying phase. They are potentially nonlinear, allowing our model to capture changes in the degree of dependence among phases as a project evolves. For example, chip designers in an application-specific integrated circuit (ASIC) project may be able to develop certain standard elements of the design (memory registers, data bus) with early information about customer requirements, but may be unable to continue until full specifications for the required functionality are released. Figure 17 shows how these constraints on the work that is available for development from within a phase and from upstream phases can limit progress.

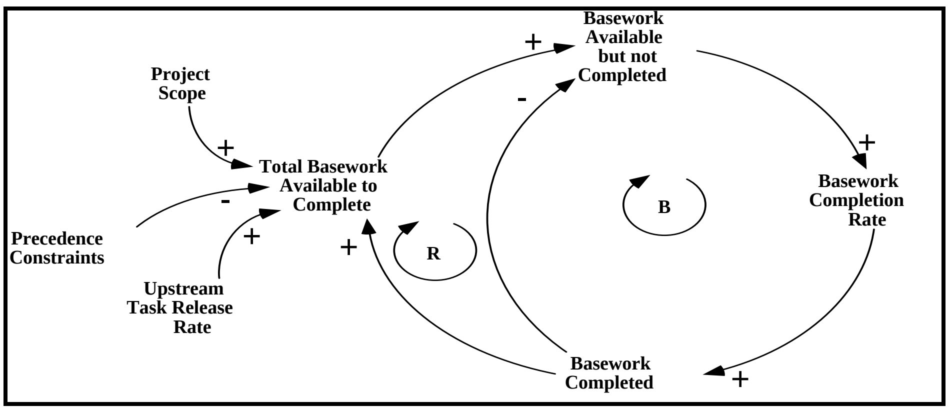


Figure 17. Development Progress Constrained by an External (between phases) Precedence Relationship

Modeling Incremental Development with Multiple Development Blocks

Figure 18 depicts an acquisition project with multiple increments or blocks. The first block is the same as Figure 15 above. Subsequent blocks have the same basic information flow, but can also be delayed by the completion of phases in previous blocks or constrained by the progress in their own blocks. Importantly, in addition to the flow of information downstream through phases (black arrows in Figure 18), multiple iteration acquisition also provides opportunities for information to flow upstream, such as from User Product Testing in an earlier iteration to Develop Requirements or Advanced Development in a subsequent iteration (red vertical arrows in Figure 18).

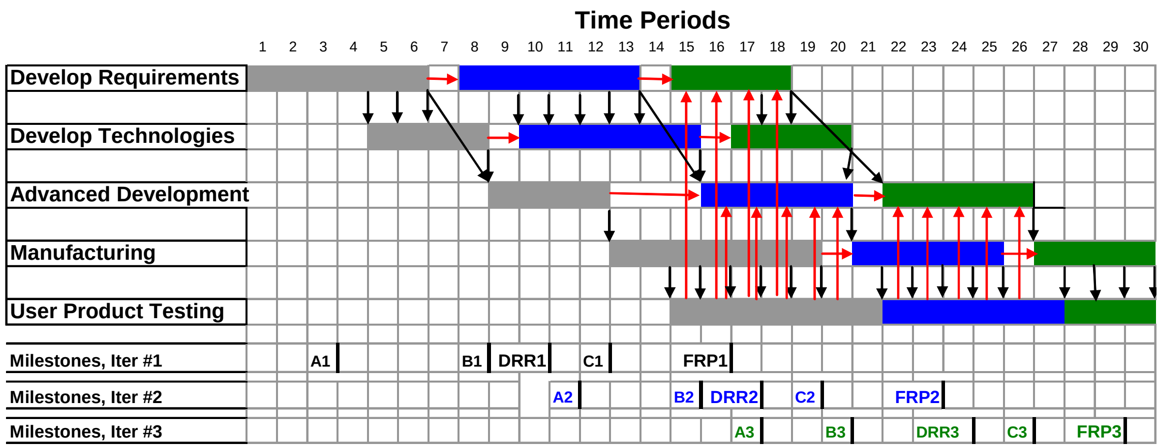


Figure 18. Information Flows in an Incremental Acquisition Project

In the model, the structure of each block is the same, although parameter values are varied to reflect different acquisition projects and strategies. For example, all phases include start-up work that is not directly applied to generating development products (requirements, technologies, component designs, or products). Each phase also includes the requisite review work that also does not directly generate product. This is consistent with GAO recommendations to manage each development block like an individual project. One impact of this loading of each phase with start-up and review work that we suspect has only been recognized informally is a significant increase in the total amount of work required to provide a given set of requirements to warfighters when multiple development blocks are used.

As will be shown with the model, this work has a significant impact on project performance that may impact the types of projects in which spiral development can be effective.

A Formal Model of Spiral Development

The conceptual model described above was used to build a formal computer simulation model of an acquisition project that can reflect traditional and incremental or spiral development strategies. The simulation model is a system of nonlinear differential equations. Each phase is represented by a generic structure, which is parameterized to reflect a specific phase of development. The unit of measure for development work is the task or work package, an atomic piece of work. Examples include writing a line of code or installing a steel beam. When work packages within a phase are heterogeneous, the unit of work can be defined as the average amount an experienced person can accomplish in a given interval. In the model, a work package is estimated to be the amount of work a developer can accomplish in a year (e.g., a person-year of work).

Modeling the Flows of Acquisition Work

The model represents workflows through a project phase as a value chain of alternating backlogs and development activities with two rework cycles (Figure 19). The value chain is described with the boxes and pipes with valves along the bottom of Figure 19. The value chain passes from the Initial Completion Backlog through the Initial Completion Rate into the Quality Assurance Backlog, through the Approval Rate into the stock of Work Approved, and through the Release Rate to the accumulation of Work Finished and Released. The rework cycle is inherent in development projects and has been modeled and used extensively to explain and improve project management (Lyneis, Cooper & Els, 2001; Ford & Sterman, 1998; Cooper & Mullen, 1993; Cooper, 1980; 1993a;b;c; 1994).

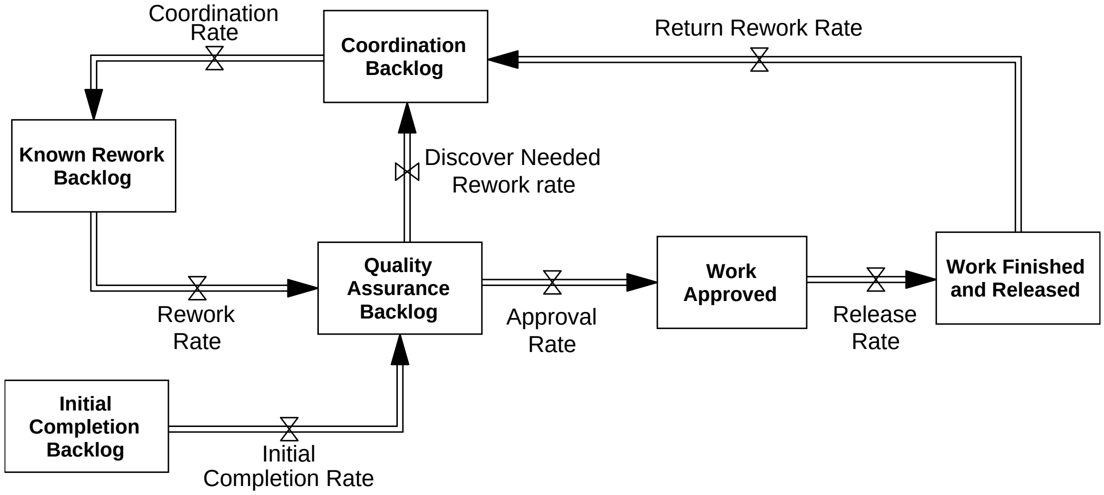


Figure 19. Work Backlogs and Flows through a Development Phase

The model used here describes the flows of work through a project in which all work starts in the backlog[27] of work needing to be initially completed ("Initial Completion Backlog," box at bottom of Figure 19). As work is first completed, it enters the stock of work needing quality assurance (QA). Quality assurance could take many forms, including reviews of designs by senior engineers, prototype building and testing, and the inspection of work. Work needing quality assurance accumulates in a Quality Assurance Backlog (box in middle of Figure 19). If work passes QA (either because it is correct or the need for changes is not detected), it is approved and adds to the stock of Work Approved. When sufficient work has been approved, a package is released, adding to the stock of Work Finished and Released to other phases or users. The release package size is a management decision, often based on the characteristics of the phase. For example, in semiconductor development, the vast majority of the design code must be completed prior to release for a prototype build since almost all the code is needed to design the masks. In other development settings, managers have broad discretion in setting release package sizes.

[27] Because the flows of development activities reflect the completion of the activity, the backlogs, as used here, include work in progress as well as work on which development has not yet been started.

Work found to require changes moves into a stock of tasks requiring changes that must be resolved through coordination with the phase responsible for the problem ("Coordination Backlog"). Classic examples include designers working with users to refine ambiguous or infeasible requirements or manufacturing engineers meeting with product designers to explain why parts can't be built as specified in the drawings. After coordination resolves disputed issues, these tasks move to the stock of work known to need rework ("Known Rework Backlog") and are subsequently reworked, then returned to quality assurance for re-inspection, testing, etc.

Quality assurance is imperfect, so some tasks requiring rework can be missed and are erroneously approved and released. These rework requirements may be discovered later by another work phase. In industry, if they are not discovered they remain embedded in the product after it is released, to be discovered by the customer. In our model of acquisition, we assume that all defects are discovered in final product testing by users. When the phase that discovers the problem reports it, the generating phase is notified, and the affected tasks are moved from the stock of work considered finished to the coordination backlog, then eventually reworked. For example, a test phase may discover a short circuit across two layers in a prototype chip. If the error is traced to the design, test engineers must notify the designers and work with them to specify the location and characteristics of the short circuit. The designers then must rework, re-check and re-release the design, followed by changes in layout, tape-out, masking, and prototype fabrication.

Given the arrangement of development activities in a phase described above, progress is constrained by the rate at which work packages move through the flows that connect the stocks. Four development activities and several development features control rates. The initial completion, quality assurance, coordination, and rework rates are each the lesser of the rate allowed by the availability of work or the resources applied (described later). The rates allowed if the development process has infinite resources (i.e., uncapacitated conditions) are described with an average processing time assuming all labor, equipment, knowledge and understanding are available. Project progress depends largely on how much work gets trapped in the rework cycle versus how much "leaks out" of the rework cycle through approval. The fraction of work discovered to require rework is used to model project complexity. More complex projects are assumed to require more iteration for completion.

Modeling Concurrence

As described, concurrence often constrains the rates and development progress. Internal precedence constraints are modeled with a (potentially nonlinear) function that relates the fraction of a phase's work that has been released to the fraction of the phase's work that is available for initial completion. For example, an internal precedence relationship in which 100% of the work was available regardless of the fraction released would reflect a development phase in which all of the work can be developed simultaneously. In contrast, an internal precedence relationship that starts at 20% of the work being available and rises steadily at a rate of 1 work package becoming available for each released until 100% of the work is available when 80% has been released would prevent more work from being initially completed if 30% of the work had been initially completed but lots of rework prevented more than 10% from being released. As examples, three internal precedence relationships from a semiconductor development project are shown in Figure 20.

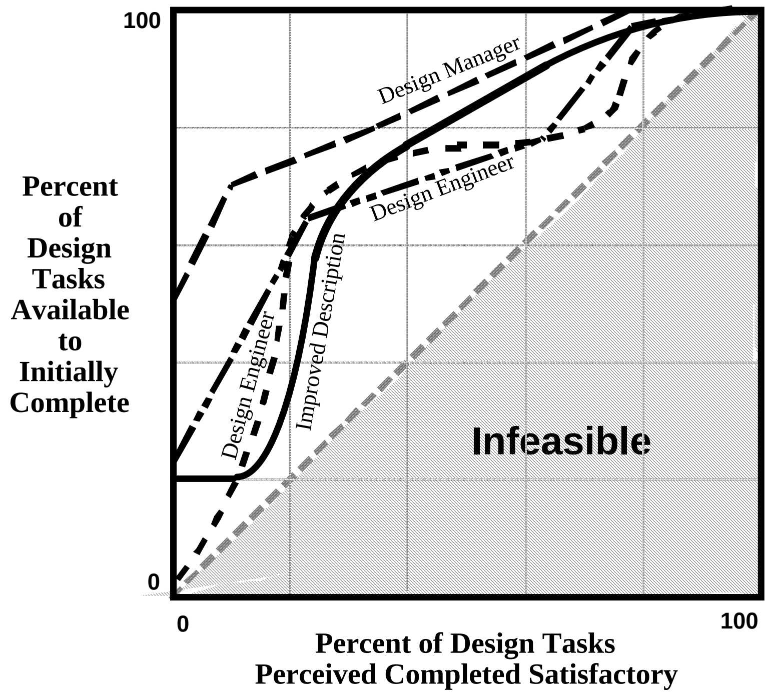


Figure 20. Modeling Concurrence—An Example of Three Internal Precedence Relationships

Like a development phase's Internal Precedence Relationship, an External Precedence Relationship between two development phases can act as a bottleneck in the availability of work. The Critical Path and PERT methods model static inter-phase dependencies in development projects and product development research (e.g., Rosenthal, 1992; Clark & Fujimoto, 1991; Eppinger, Whitney, Smith & Gebala, 1990) by specifying the temporal relationship between start and end-times of activities. The purpose of External Precedence Relationships is the same as the precedence relationships used in the Critical Path and PERT methods: to describe the dependencies of development phases on each other for the initial completion of work. However, there are several important differences between External Precedence Relationships and precedences used in the Critical Path and PERT methods.

- External Precedence Relationships describe the dependency between two phases along the entire duration of the phases, instead of only at the start and finish of the phases, as in the Critical Path and PERT methods.

- External Precedence Relationships can be nonlinear.
- External Precedence Relationships describe a dynamic relationship between development phases by allowing the output (Percent Tasks Available for Initial Completion) to fluctuate over the life of the project depending on the current conditions of the project, as described by the External Precedence Relationship's input (Percent Upstream Tasks Released).

External Precedence Relationships can be used to describe rich inter-phase relationships which cannot be described with Critical Path and PERT precedences. For example, a downstream phase which is constrained by the release of upstream tasks throughout its duration (not only at the beginning or end of the phase) in a linear relationship can be described with a "lockstep" External Precedence Relationship. Such a relationship could be one that does not make any work available until some work has started and increases the amount available steadily at 2% of the work available per percent released until all of the work is available when 50% of the upstream work has been released. External Precedence Relationships are often nonlinear, as demonstrated by the descriptions of the relationship between the product definition and design phases of a semiconductor chip project in Figure 21.

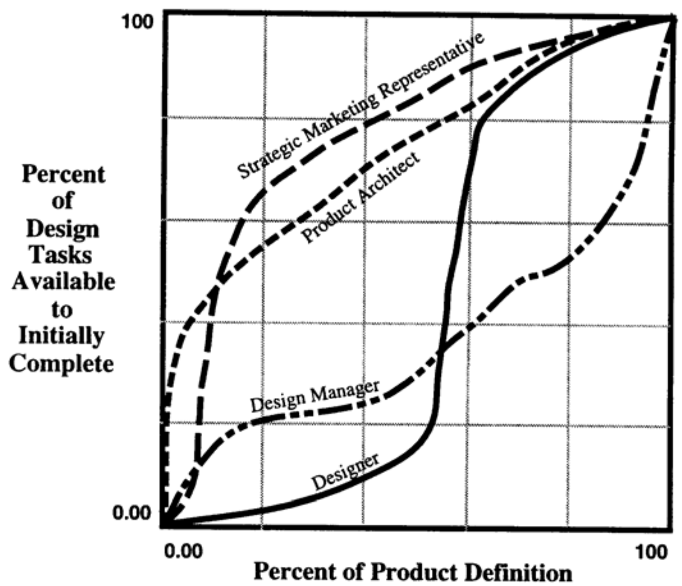


Figure 21. Modeling Concurrence—An Example of Four External Precedence Relationships

Modeling Resources

The model simulates two types of development resources. Either resource type can constrain progress by limiting the development rate. Direct resources are the people and associated equipment required to perform the development work, i.e., to develop requirements, develop technology, design products, manufacture products, and test requirement satisfaction for use. Indirect resources perform project management and associated work that support and facilitate development. Total direct resources are assumed fixed and allocated based on the backlogs of work available to be developed (the stocks represented as boxes in Figure 19). In contrast, indirect resources (also assumed fixed) serve the performance of activities (the development rates, the pipes with valves in Figure 19) and are distributed proportionately based on the size of those development activities. As will be shown, the model indicates that, when there are many development activities occurring simultaneously (e.g., in spiral development), project management (indirect) resources can constrain progress.

Resource allocation for direct and indirect resources is based on allocation fractions. Target fractions are the proportion of total indicated demand for resources generated by each activity. See Joglekar and Ford (2005) for a detailed description. The applications of allocation fraction targets are delayed to reflect the many physical and informational processes that are required. Research supports the important role of delays in controlling dynamic systems such as acquisition projects. For example, structural control system researchers have studied how delays between signals from sensors and actuators impact structural system behavior and found that purposeful time delays can improve structural behavior over eliminating time delays (Mahmoud & Al-Muthairi, 1994; Udwadia, Bremen, Kumar, & Hosseini, 2003). Allocation delays are modeled with first-order exponential adjustments that move applied allocation fractions toward targets a fixed portion of the difference between the applied and target fractions each time period (see Lee, Ford, and Joglekar, 2007 for more). The speed of adjustment is defined by this resource adjustment delay, with large delays generating slower adjustments and vice versa.

Modeling Project Performance

Project performance is measured in three dimensions: schedule, cost, and performance risk. Schedule performance is measured in the time required to have a given number or fraction of requirements tested and approved by users. Cost is measured in dollars based on the size of direct and indirect work forces and the duration of phases and blocks. Performance risk is measured with the average percent of the requirements provided (approved by users) at any given time. This average reflects the combination of multiple requirements. Some of the requirements may have binary performance, i.e., they work or they don't work. Other requirements may have discrete steps or continuous performance relative to requirements, such as weight or unit manufacturing cost. All the requirements can be considered met completely when the average percent of the requirements provided is 100% for a development block.

Model Calibration and Testing

The formal model was calibrated to the Javelin project described above. Data was collected from a project manager on the project (the first author) concerning the scope and work effort of each development phase, start-up and review-work requirements, and durations of development phases. For example, the Javelin project representatively had 30 requirements, 8 technologies to develop, about 200 components to design, and 3500 units to manufacture. User Product Testing validated the 30 requirements. These were modeled as performance units. Work packages, representing a fixed amount of effort, flow through the model. The number of work packages required to develop each performance unit was estimated using project manager estimates of the total work required in each phase.[28] Behavior data on the Javelin project was also collected. The Javelin project utilized a single development block. Developing Requirements and Developing Technologies were each estimated to take about 2 years, and Advanced Development was estimated to have taken 4.5 years. Total costs were estimated to be approximately $700million.

Model Testing

As discussed above, the model was developed as a tool to investigate the impacts of acquisition strategies, not to predict specific project performance. Therefore, consistent with the system dynamics approach, the behavior modes (shapes of behaviors over time) and how behavior modes differ with acquisition strategies is important, not exactly when changes or maximum or minimum values occur or their sizes. Therefore, the model's ability to reflect behavior should be based on its ability to show behavior modes, such as increases and decreases when they should occur and at increasing or decreasing rates of change.

System dynamics models should be exposed to a variety of tests to improve their reflection of the target system and to develop confidence in the model's

[28] See Ford and Sterman (1998) for a discussion of the use of work packages (development tasks) as units and their reflection of work effort.

usefulness for its intended purpose. Forrester and Senge (1980) suggest three types of tests of system dynamics models: structural similarity to the actual system, reasonable behavior over a wide range of input values, and behavior similarity to actual systems. Using several tests described by Sterman (2000), the model was tested for the structure's similarity to system structure, consistency, reasonableness of behavior, and similarity of model behavior to system behavior.

Basing the model on previously validated models, the literature and data collected about acquisition projects improves the model's structural similarity to actual acquisition projects as practiced. Model behavior was tested with extreme input values—such as no discovery of errors and very large resource quantities and productivities—as well as more typical conditions. Model behavior remained reasonable across wide ranges of input values, including extreme values. For example, discovering no errors reduces durations but also decreases quality. These tests increase confidence that the model generates realistic project behavior patterns due to the same causal relations found in the type of projects investigated (i.e., generates "the right behavior for the right reasons").

The model also reproduces the known system behavior. Figure 22 shows the simulated the work in each phase that has been initially completed until the phase has released all work. The vertical axis of Figure 22 and subsequent graphs labeled "Work being Developed (work packages)" can also be interpreted as the amount of work effort currently being used since work packages are proxy for work being performed.

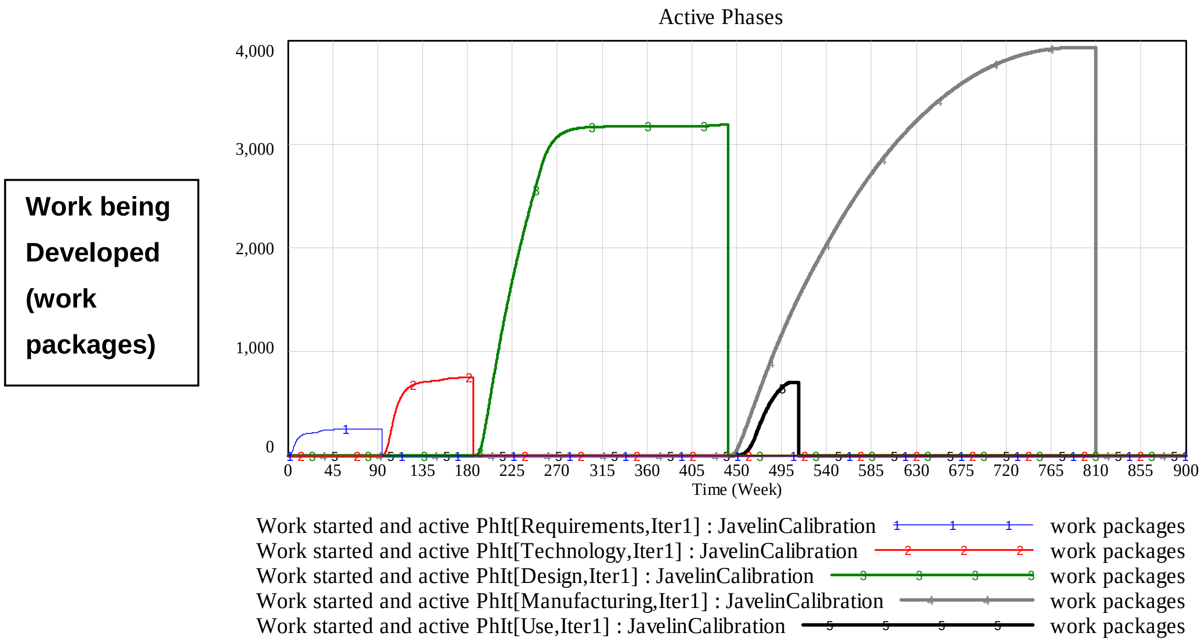


Figure 22. Test of Model Ability to Simulate Development Phases and Overlapping—Active Phases in Javelin Project

The simulated behavior of the Javelin project is consistent with the phase durations provided by the project manager, supporting the ability of the model to reflect the dynamics of the Javelin project. The simulated cost of the Javelin project ($722million) is also consistent with the data provided by the project manager, supporting the ability of the model to reflect the Javelin project cost performance.

Figure 23 shows the simulated performance risk for the Javelin project, the fraction of requirements satisfied by specific durations that can reflect deadlines. The model behavior is similar to the Javelin project, with a single testing phase of all requirements by users (one step) and the provision of all requirements (100% average percent of requirements provided).

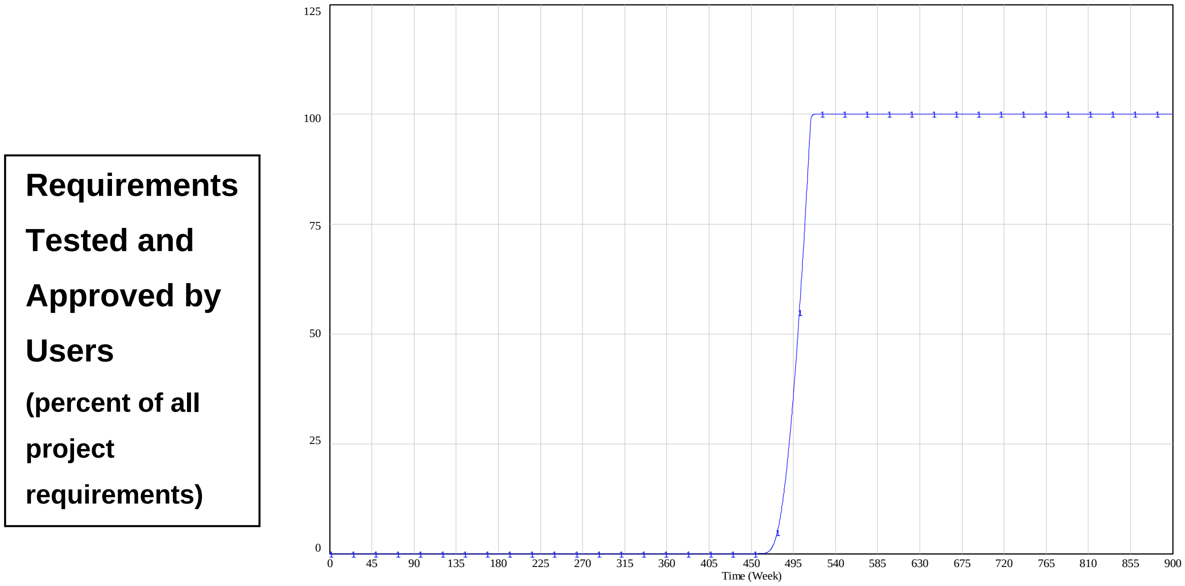


Figure 23. Simulated Satisfaction of Javelin Requirements

The model was also tested for its ability to simulate known and expected impacts of applying spiral or incremental development. If a model accurately reflects the impacts of incremental development, it should simulate that the same project with multiple development blocks provides some (but not all) requirements to users earlier, provides requirements in steps at the ends of development blocks, and probably provides all requirements later than the project if done in a single block. To test the model's ability to reflect incremental development, the model as calibrated to the actual Javelin project was changed to reflect development in three blocks. The primary management decision required to implement this change is how many of the 30 total requirements and other work to develop in each of the three blocks. For this test, it was assumed that the requirements were distributed evenly across the blocks (10 requirements per block). The scope of the other phases (e.g., new technologies, design components, and units to manufacture) were also distributed approximately evenly across development blocks.[29]

Figure 24 shows the simulated performance risk of the Javelin project as calibrated and the Javelin project as simulated in three development blocks.

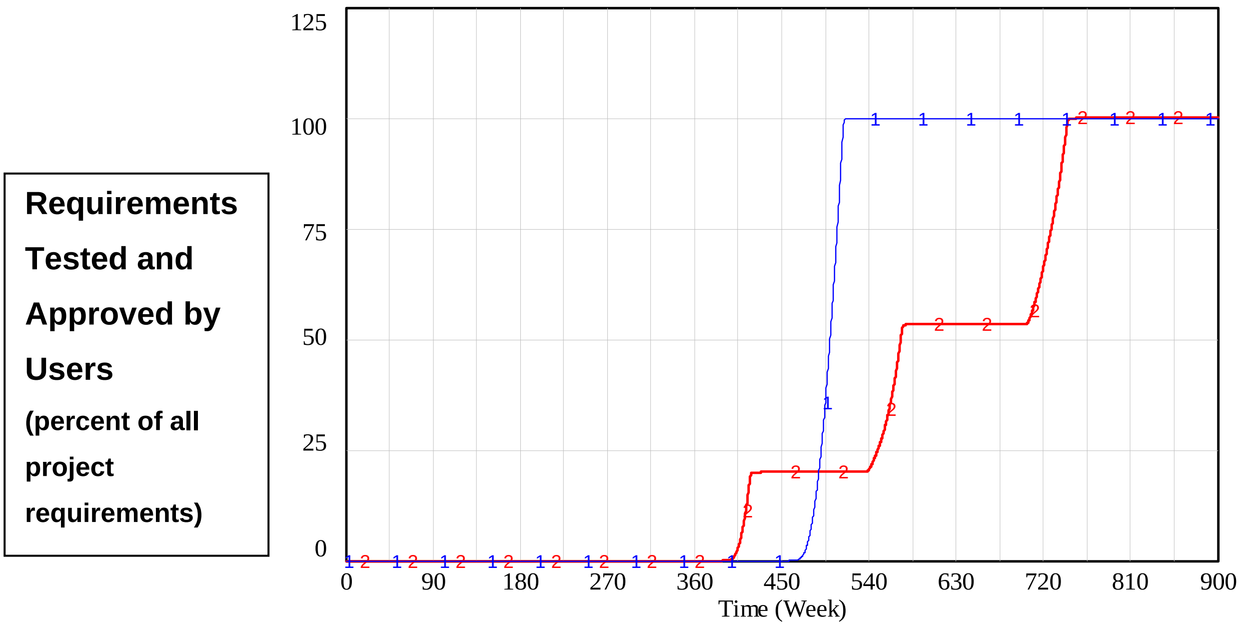


Figure 24. Test of Model Ability to Simulate Single-block and Incremental Development—Javelin Project in One (Line 1) and Three Even (Line 2) Development Blocks

The model reflects the impacts of incremental development described. When compared to a traditional approach (line 1), the incremental approach (line 2) provides some requirements earlier, satisfies requirements in steps, and satisfies all

[29] An even distribution of scope across development blocks for all phases was chosen for clarity and consistency. In actual projects, the distributions would be determined by the needs of individual blocks (e.g., which requirements need which technologies) and by the design of the project by project management.

requirements later. The simulation also supports an expected increase in cost from $722m for traditional to $1531m for spiral. The timing and sizes of the steps vary with the allocation of requirements and other work to blocks, resources and other model calibrations; but the changes in behavior mode support the model's ability to reflect differences in acquisition strategy.

As an additional test of the model, the size of the development staff was doubled for the Javelin calibration project. If the model reflects actual projects, this change should speed up development but increase costs.

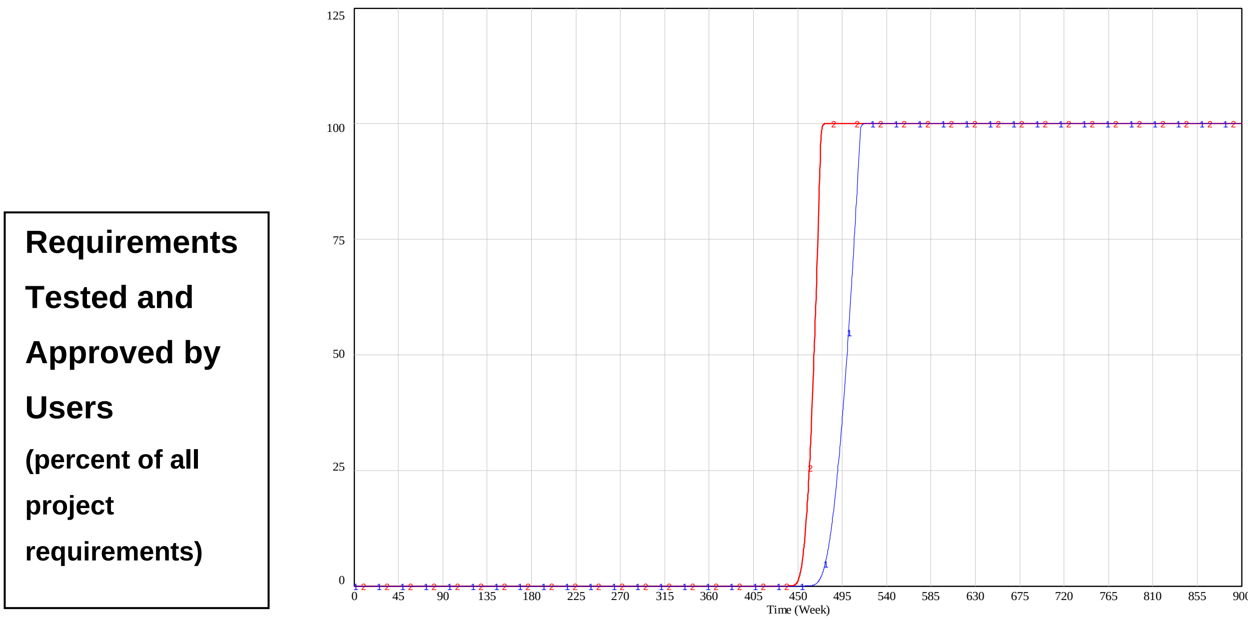


Figure 25. Test of Model Ability to Simulate Impacts of Resources on Progress—Javelin Project in One Block (Line 1) and with more developers (Line 2)

More resources generate products faster but at much higher cost. Doubling the number of developers saves 30 weeks (100% of requirements satisfied in week 491 instead of week 521) but increases costs dramatically from $722m without the larger development staff to $1,327m (an 83% increase).

Based on these and additional tests, the model is considered useful for the investigation of the impacts of acquisition strategies on project performance.

Model Use

Two focusing questions which address the issues revealed by the literature and case study portions of this report were used to guide model use:

Q1: What are the impacts of a spiral/incremental development approach compared to a traditional single-block development strategy?

Q2: How might successful spiral/incremental development project performance differ from the successful management of single-block development projects?

The Impacts of Incremental Development on Acquisition Project Performance

The first question is addressed by simulating the same project using a traditional single-block development strategy and an incremental development strategy and comparing the behavior of the two projects. As described above, the model structure includes the fundamental features that distinguish incremental development from traditional development (e.g., multiple development blocks, concurrent development blocks, additional start-up, reviews, contracting, etc.) and, therefore, can simulate behavioral and performance differences.

The calibration project case (Javelin) fully satisfied all its requirements. However, not satisfying, or partially satisfying requirements reflects the project risk and is, therefore, an important performance measure. Therefore, to facilitate the comparison of project performance using different strategies, a Base Case project was created that does not fully satisfy all requirements based on the Javelin calibration project. Figure 26 shows the Performance Risk Profile of three project simulations: 1) the calibration project (Javelin), 2) the Base Case project (Javelin without 100% satisfaction) using a single-block strategy, and 3) the Base Case project using an incremental development strategy with the requirements and work distributed evenly across three development blocks.

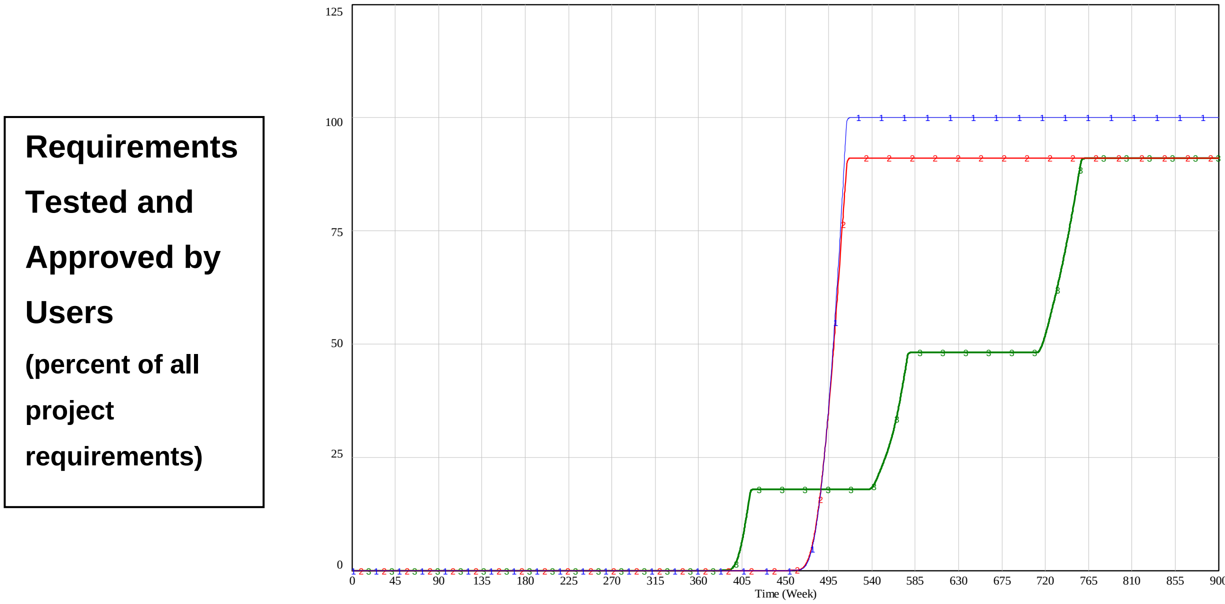


Figure 26. Performance Risk Profile of a Calibration, Base Case, and Incremental/Spiral Project

Table 1 compares the performance of these three simulated projects. The first two performance measures reflect schedule performance with the project duration required to satisfy the first requirement and the project duration required to satisfy all the requirements that the project will satisfy. The third performance measure reflects cost performance with the estimated development cost. The last two performance measures reflect project risk with the percent of the total project requirements satisfied by a specific deadline. For Table 1, the deadline was chosen to be the time when the Base Case project using the traditional strategy satisfied all the requirements the project would satisfy.

		Units of Measure	Project Scenario: Javelin	Base Case - traditional	Base Case - spiral
Performance Measure	Duration to first requirement satisfied	weeks	471	***470***	**397**
	Duration to max. requirements satisfied	weeks	520	**518**	***762***
	Total development cost	$1,000,000	722	**719**	***1,555***
	Requirements satisfied by deadline	%	**100**	91	***18***
	Final requirements satisfied	%	**100**	***91***	***91***

Table 1. Performance Comparison of Three Simulated Acquisition Projects

Although simulated values are relative and not predictions, the results in Table 1 identify important impacts of incremental/spiral development on acquisition project performance when compared to a traditional single-block strategy. Underlined bold values in Table 1 indicate the best performance among the three projects for each performance measure. Values in bold italics indicate the worst performance among the three projects for each performance measure. Notice that compared with the Base Case—traditional project, the Base Case—spiral project is best in only one performance measurement (Duration to first requirement satisfied) but is worst in three other performance measurements (Duration to max. requirements satisfied, Total development cost, and Risk—requirements satisfied by deadline). This demonstrates the ubiquitous tradeoffs in performance that different strategies present. If all performance measures were valued equally, spiral development would appear to be a poor choice as an acquisition strategy. However, not all performance measures are of equal value in all acquisition projects. ***Consistent with the case studies and analysis above, these model results identify the one performance measure that must be most important for a spiral***

development strategy to improve total project performance—Duration to first requirement satisfied.

Causal Analysis and Explanations of Model Behavior

Analyses of the structure of the model provide a means of explaining the results shown in Figure 26 and Table 1, i.e., why spiral development changes project performance the way it does. Here also lies an important definitional distinction: we use the term spirals and increments here somewhat interchangeably, since all spirals eventually become defined. But in precise terms, our model results here refer specifically to the effects of *deliberate deferral of work to successive increments*, versus the unplanned, inestimable and open-ended nature of true spiral development. To identify the causes of specific behaviors, the behavior of specific model variables is traced through the causal pathways in the model from a performance variable "backwards" up the causal pathway to reveal the drivers of, and constraints on, performance. For example, schedule performance is constrained by the progress rates of different blocks and phases, which can be constrained by either the availability of work or progress rates allowed by resources (the model structure analysis identifies which constrains progress). The availability of work can be constrained by the completion of upstream work or the amount of work remaining to be developed (again, model structure analysis reveals which controls). Resource rates can be constrained by either the quantity and productivity of developers or the quantity and productivity of project managers. Following the driving or constraining causal pathway through the model for the behavior of a specific performance variable for a specific simulation can reveal the locations of *bottlenecks*. The results of model structure analysis for each performance measure in Table 1 will be described in turn.

Model structure analysis reveals that the "Duration to the first requirement satisfied" values are constrained by the time required to get the requirements and other development products in the first block through the development phases and tested by users. This is constrained by the time taken in each phase before the development products are released to downstream phases. These phase durations are driven primarily by the progress rate, which is effected by the quantity and productivity of developers and the amount of work in each phase. Therefore, when the number of requirements and, therefore, work is reduced in the first development block of a spiral strategy, the block can be completed faster—satisfying the requirements in that block earlier.[30] This explains why the Base Case: spiral project performs best in this performance measure. A shorthand description of this causal path from this performance variable through the project structure is: Duration to first requirement—end block 1—block 1 phase durations—block 1 work required—scope of block 1. A reasonable question that model structure analysis (and more simulations) can address is, "How much faster can spiral development satisfy requirements?" Further reductions in the number of requirements in the first block reduce the duration to the first requirement satisfied, *but not proportionate to the reduction of requirements and only to a minimum duration.* This is because developer progress rates are not the only project feature that constrains progress, i.e., are not the only potential bottleneck. In this case, concurrent development also increases project management needs, and project management resources begin to constrain progress at some point. In addition, available work constraints (i.e., development processes) have minimum durations and prevent the very early satisfaction of requirements. This illustrates the important role of multiple and dynamic progress bottlenecks.

Model structure analysis reveals that the "Duration to maximum requirements satisfied" values are controlled by when the last requirement is satisfied, which is at the end of block 1 in the Base Case: traditional project and the end of block 3 of the Base Case: spiral project. In the Base Case: traditional project, this is controlled by the progress and concurrence of the phases. The progress is sometimes

---

[30] Note that if the reduction in the number of requirements in the first block was not accompanied by a reduction in the scope of the other phases in the first block, as suggested in the previous footnote, that the bottleneck in the first phase might not be addressed, and the improved performance might not materialize.

constrained at some times by resources and at other times by processes. For example, the early portion of the requirements phase does not progress faster because of the number or productivity of developers, but later in the same phase the existing developers run out of work due to the process constraints of waiting for rework to be completed and errors to be discovered. The shifting of progress constraints illustrates the importance of understanding progress bottlenecks to successfully managing acquisition project dynamics. Considering the spiral project, process constraints such as the sequential development of requirements in separate blocks prevents the beginning of the requirements phase in the last block of the incremental/spiral development project until the requirements phases in the first two blocks are completed. This forces the final block to start relatively late (over three years into the project). This late start forces the third block to compete for project management and support resources with the first two blocks, which are in progress. Direct resources (developers) constrain the progress of the phases in block 3 and process constraints such as the sequential nature of the phases set a minimum duration for Block 3.[31] A shorthand description of this causal path from this performance variable through the project structure is: Duration to maximum requirements—end last block—start of last requirements phase and [last block duration]—end of preceding requirements phase and [last block concurrence and direct resources]. The square brackets indicate a split in the causal pathway; i.e., that two paths constrain the end of the last block.

Model structure analysis reveals that the "Total development cost" values are driven by the duration that the two types of resources, the development workforce and the project management workforce, are charged to the project (labor rates are assumed to include other expenses). These workforces are fully allocated to development or project management as long as they are needed (i.e., there are

[31] The impact of the sequential phases illustrates the benefits of concurrent development. See Ford and Sterman (2003a; 2003b) for studies of the side effects of concurrent development that can limit or decay progress.

backlogs of work for the development workforce and development activities for the project management workforce). Therefore, costs are directly related to the duration of blocks and the project. Longer projects cost more. However, the driver of this total duration is the total amount of work to be completed. This consists of two types of work: work required to develop products (requirements, technologies, designs, products, test results), and indirect work to fulfill review, contracting, start-up, and other functions that are related to development phases. The more phases a project has, the more indirect work it must complete. Therefore, more development blocks increase indirect work, thereby increasing the project duration and costs. This explains why the Base Case: spiral project, which has more development blocks and phases than the other projects, has the largest cost. A shorthand description of this causal path from this performance variable through the project structure is: Total cost—2 workforces—backlogs and activities—work required—start-up, reviews, etc. work—number of phases—number of blocks.

Model structure analysis reveals that the “Requirements satisfied by deadline” values are driven by the satisfaction of requirements and the deadline chosen. For a given deadline, this performance measure depends on the progress of development blocks (described above) and, in the spiral development case, the number of requirements in each block (a project-planning decision). The dependence on the sizes of the blocks is particular to the spiral project because the structure of spiral development generates significant times of no increases in requirements satisfied. If one of these plateaus in final performance occurs at the deadline, the spiral project remains at a relatively low performance level. This is illustrated in Figure 26. This explains why the Base Case: spiral project has such a poor performance for this metric (Table 1). A shorthand description of this causal path from this performance variable through the project structure is: Requirements satisfied by deadline—progress of blocks and [sizes of blocks]—backlogs and activities—work required—start-up, reviews, etc., work—number of phases—number of blocks.

Model structure analysis reveals that the “Final requirements satisfied” values are driven by the total fraction of the requirements that pass testing by users. The model assumes that the users find all failures of the product to fully satisfy the requirements. Therefore, the defects found by users that limit the final requirements satisfied are those inherited by the user-testing phase from upstream phases. Three features determine the number of defects passed on to downstream phases and eventually to user testing: 1) the number of defects generated within a phase (e.g., a technology that cannot satisfy a requirement even if developed optimally), 2) the fraction of those defects not discovered and passed on to downstream phases (accidentally or purposeful[32]), and 3) the sensitivity of downstream phases to inherited upstream errors.[33] More errors generated and passed on and more sensitivity to those errors degrades performance in this dimension. Because inherited errors generate more errors in the downstream phases, the effects are multiplicative and grow with delays in error discovery and correction. These features are often driven by the technological relations among requirements, technologies, and design components. However, they also can be influenced by managerial actions such as quality assurance policies and developer morale. The model assumes (for simplicity) that changing to a spiral approach does not change these factors. This explains why the Base Case: spiral project and Base Case: traditional project have the same performance. If the spiral project were to cause changes in these three features (e.g., an increase in errors generated due to more process complexity caused by concurrence), the performance would change. A shorthand description of this causal path from this performance variable through the project structure is: Final requirements satisfied—two workforces—backlogs and activities—work required—start-up, reviews, etc., work—number of phases—number of blocks.

[32] See Ford and Sterman (2003a) for descriptions and analysis of the rational and purposeful hiding of known defects by qualified, well-intentioned project managers.

[33] See Krishnan and Eppinger (1995) for a model of inter-phase sensitivity to changes in designs.

Investigating Incremental/Spiral Development Management

The second research question focuses on the management of incremental or spiral development (terms used interchangeably here) projects: How can spiral development project performance be improved? A first step in improving the management of spiral development is to understand the managerial implications of spiral development. The graphics in Figure 27 show the active development phases of the Base Case project using a single development block (top) and spiral development (bottom).

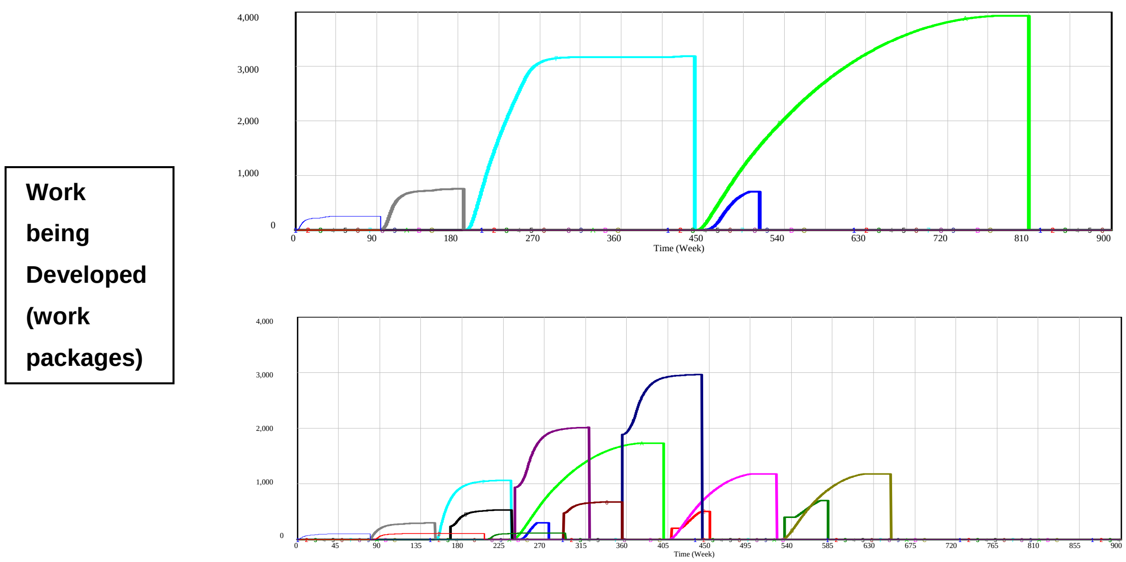


Figure 27. Active Development Phases using Single-block and Spiral Development—the Base Case Project

Phases must be coordinated with external stakeholders and other development phases. Each pair of concurrent phases creates a potential interface that requires coordination. Figure 28 shows an estimate of the phase interfaces that must be managed based on the number of active phases shown in the previous figure.

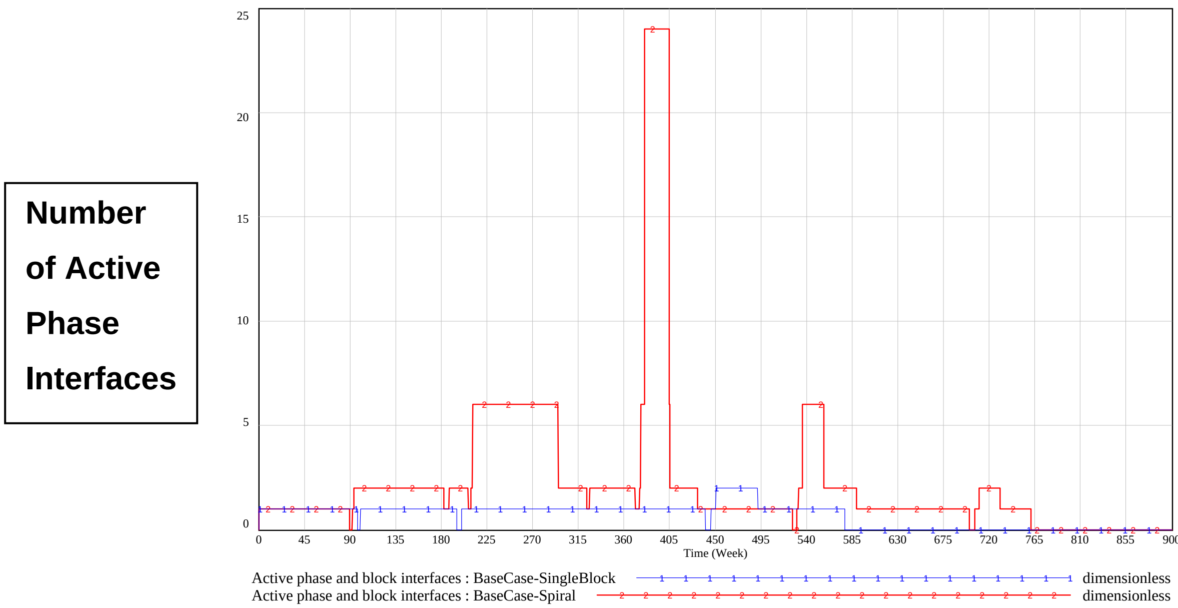


Figure 28. Performance Risk Profile of a Calibration, Base Case, and Spiral Project

Although the number of interfaces with external stakeholders and between development phases is project-specific, the impact of spiral development on project management requirements is clear. Spiral development requires significantly more coordination than single-block development.

The Critical Role of Progress Bottlenecks

Bottlenecks that constrain development progress can be caused by several different parts of a development project and located in many places. Understanding and managing them effectively is critical to successful spiral development project success. This can be illustrated by simulating projects using spiral development with different amounts of resources—a common project-management tool. The Javelin Project was simulated assuming four conditions:

1. a single-block approach (blue, Line 1 in Figure BBBB),
2. with a spiral approach (red, Line 2, in Figure BBBB),

3. with spiral and additional developers (green, Line 3 in Figure BBBB),

4. with spiral with additional developers and additional project management (grey, Line 4 in Figure BBBB).

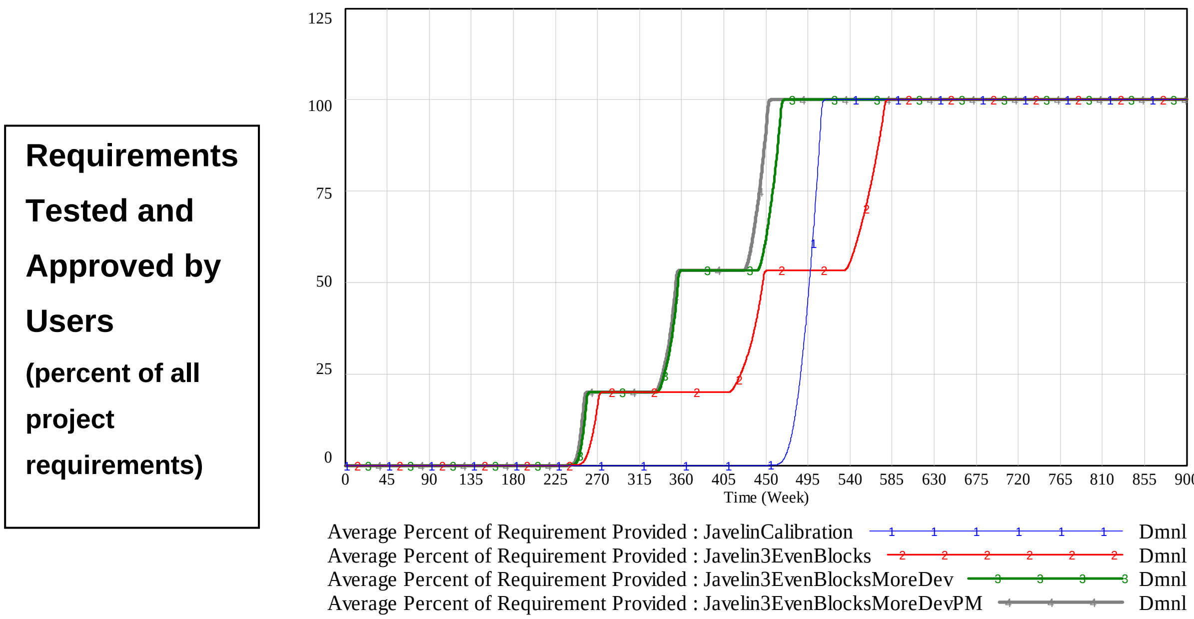


Figure 29. Different Impacts of Adding Resources on Performance—Javelin Project with More Developers and Project Management

Adding developers reduces the duration of block 2 (second and third steps are earlier), but not does not significantly change the first increment. This is because the first increment is constrained by process with significantly fewer developers than the second development block. This illustrates the importance of identifying and understanding the progress bottleneck. In this case, adding developers does not significantly reduce the first development block and would not be a very effective policy (or use of resources) if a project manager was attempting to speed up the time to First Unit Equipped with the requirements in the first block. ***Adding resources where they do not relax a progress constraint does not improve performance (an old lesson). Knowing where what project features constrain progress is particularly difficult in incremental/spiral development because of the increased project dynamics (a new lesson).*** In contrast, adding developers improves performance if the management objective was to speed the time to the First Unit Equipped with requirements from the second block. *Again, knowing where what project features constrain progress is critical for improving spiral development performance.*

The addition of project management in addition to developers (line 4 in Figure 29) also illustrates the challenges and importance of identifying and understanding progress bottlenecks in spiral development. This only impacts the third development block. This is because, in the model as calibrated, the first two development blocks have adequate project management; therefore, adding more project management does not improve performance. In contrast, the third development block is (at least partially) constrained by project-management resources, and benefits from adding more project management. In this case, the location of the bottleneck shifts from developers to project managers and is different in different development blocks. The fundamental lesson from the model is the same: ***Understanding the location of progress bottlenecks is particularly difficult but vital for successful spiral development management.***

Of additional interest, the estimated costs of the four simulated Javelin projects shown in Figure 29 are:

1. Single block: $704million
2. Spiral: $939million
3. Spiral with additional developers: $1,761million
4. Spiral with additional developers and project management: $1,753million

The first increase in cost from a single-block development ($704m) to a spiral development ($939m) is expected and has been discussed above. The second increase in cost from spiral development ($939m) to spiral development with more developers ($1,761m) is also expected and is due to the larger workforce. However, the decrease from spiral with more developers ($1,761m) to spiral with more developers and more project management ($1,753m) is counterintuitive. How can ***adding*** more resources (project management) ***decrease*** project costs? A causal path analysis of the model structure reveals that when project management resources constrain progress, adding those resources can reduce project duration, allowing an earlier release of the (expensive) developers from the project. Without the additional project management, some developers are unable to be fully utilized due to project management issues that are not being addressed. The additional project management relaxed that progress bottleneck, thereby allowing improved use of developers, faster completion of the project, and reduced costs. ***The counter-intuitive cost behavior of these simulated projects illustrates the challenges and importance of identifying and understanding progress bottlenecks in spiral development projects.***

Simulation Modeling Results Summary

The simulation model was used to investigate the impacts of spiral development on acquisition projects and the management of spiral development from a causal-path perspective. Spiral development was found to have several important impacts on acquisition projects when compared to a traditional single-block development approach. Ceteris paribus (all other things held constant or equal), the model found, or supported other findings of, the following impacts:

- Incremental/Spiral development can provide the First Unit Equipped with some (but not all) requirements satisfied faster than single-block development
- Incremental/Spiral development provides satisfied requirements to users in multiple steps or increments, whereas single-block development satisfies all requirements in a single step
- Incremental/Spiral development takes more time to satisfy all requirements than single-block development
- Incremental/Spiral development costs more than single-block development to satisfy the same requirements

- Incremental/Spiral development has a high risk of not satisfying all requirements by the time single-block development can satisfy all requirements

- The causal paths that drive and constrain project performance in incremental/spiral development pass through multiple types of resources, development processes, and move across both development phases and development blocks. The causal paths vary widely for different performance measures. This makes the drivers of and constraints on spiral acquisition project performance more difficult to identify than those influencing single-block development projects

These results indicate that incremental/spiral development is a significantly different approach to acquisition than single-block development; therefore, it requires different planning, resourcing, and management.

The model was also used to investigate the management of spiral development when compared to traditional development. Spiral development was found to have several significant impacts on acquisition project management. Investigations with the model found that (ceteris paribus):

- The concurrent use of multiple development blocks in spiral development significantly increases the number of development phases and activities that must be managed and coordinated at any given time compared to single-block development. This increases the project management needs for successful acquisition in spiral development projects when compared to single-block projects.

- Like in single-block development, progress in spiral development requires the identification and understanding of progress bottlenecks. However, the concurrence and resulting complexity of development in spiral projects causes the types and locations of bottlenecks to vary widely and be more difficult to identify and address than those in single-block development.

- Causal paths of the drivers and constraints on project performance and progress bottlenecks move from one feature of a project to another as projects evolve. The increased dynamics of development in spiral development projects when compared to single-block development make identifying and addressing causal paths and progress bottlenecks more difficult.

- Progress bottlenecks can cause counterintuitive behavior, such as reductions in project cost by adding resources at a bottleneck. Understanding and exploiting the opportunities provided by these behaviors requires a deep understanding of the project structures and dynamic interactions that drive and constrain progress.

These results indicate that incremental/spiral development requires more, different, and more difficult project management than single-block development that focuses on the identification and management of causal paths and progress bottlenecks based on the structure of the development project.

THIS PAGE INTENTIONALLY LEFT BLANK

Balancing Risks with Development Approaches

In 2004, Barry Boehm, creator of the spiral development model, released a book about software development entitled, *Balancing Agility and Discipline*. In this pragmatic book, he says that two opposing and conflicting methodologies have emerged in the software domain—that of traditional, plan-driven, processed-based (*disciplined*) and that of rapid change and adaptability (*agile*). Proponents of each of these software development approaches have their line of reasoning. The traditionalists value consistency of processes, exemplified within the Software-Capability Maturity Model (SW-CMM), and emphasize proper documentation to provide history and a knowledge base of experience. The agilists value rapid response to change versus following plans and functional software over comprehensive documentation. Disciplined methods are systematic and predictable, but can become bureaucratic as quality-oriented and risk averse. Agile methods are dexterous, but can become ad hoc and chaotic. Both value quality, but from differing viewpoints. Where the SW-CMM defines quality as specification and process compliance, agile methods view it as customer satisfaction. He asserts that the perplexing dilemma for project managers is the need for both *coping with change* and *retaining control*—since both approaches have their advantages and drawbacks.

The two approaches have evolved over the past three decades and are still changing:

Disciplined methods	The plan-driven, disciplined approach emerged from systems engineering and quality disciplines because of the growing complexity of large aerospace programs. Software, as an essential but physically unconstrained component, grew to need "disciplining" via standards and structured techniques within a requirements/design/build paradigm. This gave rise to standards and repeatable processes, emphasis upon defined system architecture, verification and validation, and an analytical approach to identify and manage potential risks.

Agile methods	The agile approach grew out of demands for faster product cycle-time, rapid prototyping experiences, and a philosophy favoring human interaction and flexibility versus mechanistic methods. Agile concepts are embracing informality, change, simplicity, many and frequent product releases, and "bare sufficiency" (addressing only high-priority functions).

While Boehm describes evolutionary and incremental processes being used in both approaches, the DoD's spiral development approach seems most analogous to Boehm's agile methods. And Boehm states his own, "skepticism that pure agile methods can be used effectively with large, complex, or safety-critical software systems" (Boehm & Turner, 2004). He also attributes "over-responding to change" as causal "for the $3 billion overrun of the Federal Aviation Administration's Advanced Automation System for national air traffic control" (2004). He conveys that agile methods are more risky, stating, "the *necessity of discipline to ground adaptability* is as necessary as it has ever been, especially as system software size and complexity grow" (2004).

But also clear are the benefits of each of Boehm's competing approaches. Discipline is needed as a control mechanism to avoid risk, but agility is needed to respond quickly to customer needs. He warns against the misuse and universal application of either, saying, "One size fits all is a myth" (Boehm & Turner, 2004) And he advocates a balanced approach between use of both methods—based upon cost, schedule and technical performance risk. In addition to organizational culture and developer personnel qualifications, he actually advocates the more disciplined, risk-averse approaches for projects that are mission/safety critical, larger in size, and have more stable requirements.

We believe Boehm's constructs about agile and disciplined software development methods correlate well with other, non-software product development strategies—especially with their regard to product characteristics and risk. Hardware is not as malleable as software, and also (unlike software) can be quite costly in production.

While Boehm suggests balancing agile and disciplined software development methods, we suggest there is also a need for balance within DoD's evolutionary acquisition methodologies: the balancing of project-control measures oriented against risks. Since both controls and risks have associated costs, the balance has long been conceptualized as in Figure 30 below (Wysocki, 2003).

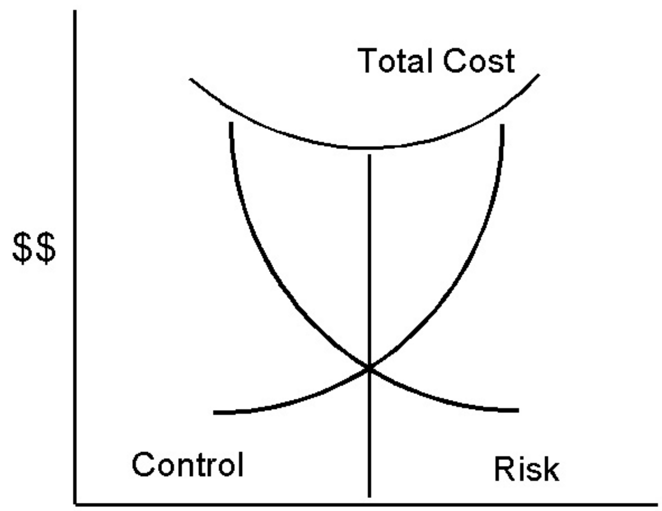


Figure 30. Perceived Relationships Among Project Cost, Control and Risk

(adapted from Wysocki, 2003)

Typical project goals are stability, discipline, simplicity and equilibrium. Program managers want these aspects with regard to program requirements, funding, design, and production configuration. But stakeholders often want flexibility, agility, adaptability and variety, and these bring about opposing tensions from change, complexity and risk.

THIS PAGE INTENTIONALLY LEFT BLANK

Conclusions

It can be summarized that spiral development was at its inception, and is at its extension by the DoD, all about risk. Paradoxically, it is an agile method envisioned to reduce risk, but which potentially can add its own. On the one hand, a spiral or incremental approach allays risk by reducing scope to render only the highest priority capabilities with the exclusive use of mature technology, and obtains early and continuous feedback from the environment for follow-on developments. On the other hand, it introduces concurrency during advanced development and adds variety in production, with all their attendant management challenges.

Although today's policy of evolutionary acquisition is prescribed as a development methodology, it is actually focused more upon what—not how—we develop. As such, it is about doable scope, reducing risk via exclusive use of mature technology. The Cost As an Independent Variable and other requirement-limiting initiatives were earlier attempts to accomplish this by encouraging product-performance trades to keep cost estimates fixed. As with CAIV, this likely means trading performance requirements for earliest deploying increments.

Spiral development also seeks to spread out the technical risk over more development and process time via incrementing. We have shown with simulation that this can potentially improve risk-management performance initially, but with higher overall costs and longer subsequent development durations, if deliberately deferring known, estimable work. As such, our computational modeling indicates that incremental development costs more and requires more time to provide the same requirements than single-step development. With regard to project risk, the increased complexity in a project using an incremental or spiral approach makes the isolation and effective management of progress bottlenecks more difficult than in single-step development.

The policy change is that spiral development now includes undefinitized increments and prescribes incremental development instead of single-step development. All amorphous spirals will eventually become defined increments—mini-programs. In years past, they have often been implemented as sequential, separate, and successive product upgrades (such as the CH-47, UH-60, C-130, B-52 program examples). But current policy expresses these as more concurrent, frequent and continuous. Such concurrency adds complexity to development models, with attendant risks of over allocation of work, noise, error, duplicity, and other inefficiencies from work deferral and divided effort in project-management organizations. Additional oversight, reviews, contracting, testing, etc., will also likely affect transaction costs. If all requirements are known and an incremental approach is used, then there is a deliberate deferral of work to later increments, and there will be a resultant increase in total development costs and durations for these same reasons.

We've suggested that a one-size-fits-all methodology for DoD system development may not be appropriate and have offered for consideration several product attributes that might help determine the efficacy of the spiral approach. We further suggest that spiral development may serve better than single-step development for initial capability when products are mutable, time critical, non-maintenance intensive, and have continuous (vs. binary) or uncertain requirements, short cycle-times (less knock-on effects), sequentially phased development, and modular independence. In contrast, spiral development may not be appropriate when there are safety or man-rating concerns and have attributes opposite to those above. In particular, PMs should understand the nature of their product requirements with regard to their range of attainment and relative to key parameters of capability, and vis-à-vis the readiness level of their enabling technologies. Some key features may indeed be binary, and others may have significant ramifications of partial attainment—such as propagated change across the entire product componentry (as in weight reduction) versus a more independent modular modification.

Open design standards will not always be incorporable, and product variety will emerge, with and without backward compatibility, interoperability, etc. Variety is both an asset (for end-users) and a liability (for manufacturers, owners and supporters). As such, to compensate for product variety, “acquirers” must “own” the design and emphasize configuration management, keeping or assigning responsibility for that function and maintaining *accountability* for it.

Our title, “From Amorphous to Defined,” alludes to both *product specification* as well as *risk realization* in spiral development. Spiral development has inherent challenges, both strategic and tactical, of which PMs must be aware. We’ve highlighted and illustrated them here, as well as have shown that spiral development can indeed work—especially for technically mature and mutable products with open or elegant architecture.

Program Managers must be aware of these inherent risks and take necessary precautions to balance them with increased use of tools, such as technology readiness levels, configuration management, technical performance measurement, contract incentives, options and phasing, organizational design, etc.

Stability is the quest in all things programmatic—for funding, requirements, design, production configuration, etc. But in an unstable world, and with the future being necessarily uncertain, the tension between control and change is probably unending. PMs do have some tools for coping, and being forewarned is being forearmed. PMs are used to concurrency and change, as they are largely what make project management what it is: a balancing act. Mechanisms for control of risk include project management tools such as configuration management, technical performance measurement, earned-value management, risk management, real options, etc. Organizational and cultural factors such as leadership, trust and accountability play a significant role as well (Zolin & Dillard, 2005, May). Successful use of these tools to balance control and risk in projects with a high rate of change and concurrency is an area for our further study.

Recommendations for Practice:

1. Project managers need to be aware of the inherent risks of spiral development and take necessary precautions to balance those risks. Many tools and control measures are currently developed and available to assist project managers in balancing the risks of spiral development, such as technology readiness levels, configuration management, technology performance management, real options, project phasing, risk management, earned value management and organizational design.

2. Incremental and spiral development projects provide additional *opportunities* for managing development risks that are inherent in the project design. These include project planning decisions about the number and concurrency of development blocks, and the requirements and associated technologies and design components to be included in specific blocks. This planning provides opportunities to anticipate where critical progress bottlenecks may occur and design how to best monitor and respond to them.

3. Product attributes may help determine the suitability of spiral development. PMs should consider such characteristics as: mutability, time criticality, man-rating, modular interdependency, key parameters of capability versus range of requirement attainment (i.e. binary vs. continuous), and the relative amount of concurrency among increments.

4. Progress bottlenecks in incremental and spiral development often oscillate between process constraints (e.g. availability of work due to upstream progress) and resource constraints (e.g. developer or project management quantities or productivities). Successfully addressing a constraining progress bottleneck often shifts the progress constraint to a different location in the project. Therefore, a structured and interdisciplinary practice of identifying and addressing bottlenecks can improve performance.

5. Configuration management accountability must be assigned and kept to maintain supportability, failure mode identification and causality and prevent the variety generated by evolutionary acquisition from reducing total product performance.

References

Abdel-Hamid, T. (1988) Understanding the “90% syndrome” in software project management: A simulation-based case study. *The Journal of Systems and Software*, 8, 319-330.

Agre, P. E. (2003). Hierarchy and history in Simon's "Architecture of complexity." *Journal of the Learning Sciences,* 12(3), Lawrence Erlbaum Associates.

Air Force Institute of Technology. (2007). Retrieved from http://www.afit.edu/ls/knowledge/KO-04/DOD5000_KO04.htm#

Apte, A. (2005, June 30). Spiral development: A perspective (NPS-AM 05-0111). Monterey, CA: Naval Postgraduate School.

Army TACMS Project Office. (1993, May 14). *Lessons learned from propulsion system nozzle failure on Army TACMS*. Redstone Arsenal, AL: author.

Army Times. (2007, February 12). Is basic training too soft? Springfield, VA.

Ashby, W. R. (1960). *An introduction to cybernetics*. London: Chapman & Hall.

Augustine, N. R. (1997, June). *Augustine’s laws* (6th ed.). LOCATION: American Institute of Aeronautics & Ast.

Boehm, B. (1985). *A Spiral Model Of Software Development And Enhancement*. ISPW: 22-42.

Boehm, B., & Hansen, W.J. (2000, February 9). Spiral development: Experience, principles, and refinements.

Boehm, B. & Turner, R. (2004). *Balancing agility and discipline*. Boston, MA: Addison-Wesley.

Boudreau, M. (2006). *Acoustic Rapid COTS Insertion: A Case Study in Spiral Development*, Monterey, CA: Naval Postgraduate School.

Bruce, M., & Cooper, R. (2000). *Creative product design.* West Sussex, England: Wiley & Sons.

Bruns, A. (2003, July 30). Presentation at Midwest Gateway Chapter of INCOSE. St Louis, MO. Retrieved from https://acc.dau.mil/CommunityBrowser.aspx?id=17651

Chairman of the Joint Chiefs of Staff. (2003, June 24). *Joint capabilities integration and development system* (CJCS 3170.01C). Washington, DC: author.

Chairman of the Joint Chiefs of Staff. (2005, May 11). *Joint capabilities integration and development system* (CJCSI 3170.01E). Washington, DC: author.

Clark, K. B., & Fujimoto, T. (1991). *Product development performance, strategy, organization, and management in the world auto industry*. Boston, MA: Harvard Business School Press.

Cooper, K. G. (1980). Naval ship production: a claim settled and a framework built. *Interfaces*, 10(6), 20-36.

Cooper, K. G. (1993a). The rework cycle: why projects are mismanaged. *PM network*, PMI, February, 5-7.

Cooper, K. G. (1993b). The rework cycle: how it really works…and reworks. *PM network*. PMI, February, 25-28.

Cooper, K. G. (1993c). The rework cycle: benchmarks for the project managers. *Project Management Journal*, PMI, 14(1), 17-21.

Cooper, K. G. (1994). The $2,000 hour: How managers influence project performance through the rework cycle. *Project Management Journal*, PMI, 15(1), 11-24.

Cooper, K. G., & Mullen, T. (1993). Swords and plowshares: the rework cycles of defense and commercial software development projects. *American Programmer*, 6(5), 41-51.

Darwin, C. (1859). *Origin of the species.* Cerf & Klopper. New York.

Defense Acquisition University. (2001, January). *Glossary—Defense acquisition acronyms and terms* (10th ed.).

Defense Acquisition University. (2003, June). *Program managers toolkit* (13th ed.). (Ver 1.0).

Department of Defense. (2004, May 12). *Model designation of military aerospace vehicles* (DoD 4120.15-L). Washington, DC: author.

Department of Defense Open Systems Joint Task Force. (2003, August). *A modular open systems approach to acquisition, program managers guide*. (Ver. 1.1). Retrieved from www.acq.osd.mil/osjtf 14 Jan 2007.

Department of Energy. (2002, August). *United Parcel Service (UPS) CNG truck fleet: Final results*. Washington, DC: author.

Dillard, J., & Nissen, M.E. (2007). Computational modeling of project organizations under stress. *Project Management Journal*. 38(1), 5-20.

Dillard, J.T. (2005). Toward centralized control of defense acquisition programs. *Defense Acquisition Review Journal*, Aug-Nov, 330-344.

Duvall, M., & Bartholomew, D. (2007). *The promise and peril of PLM*. Baseline. 5 Feb.

Eppinger, S.D., Whitney, D.E., Smith, R.P., & Gebala, D.A. (1994). A model-based method for organising tasks in product development. *Research in Engineering Design,* 6(1) 1-13.

Ford, D., & Sterman, J. (1998, Spring). Modeling dynamic development processes. *System Dynamics Review*, 14(1), 31-68.

Ford, D., & Sterman, J. (2003a, September). The liar's club: Impacts of concealment in concurrent development projects. *Concurrent Engineering Research and Applications*, 111(3), 211-219.

Ford, D., & Sterman, J. (2003b, September). Overcoming the 90% syndrome: Iteration management in concurrent development projects. *Concurrent Engineering Research and Applications*, 111(3), 177-186.

Forrester, J. W. (1961). *Industrial dynamics*. Cambridge, MA: MIT Press.

Forrester, J., & Senge, P. (1980). Tests for building confidence in system dynamics models. *TIMS Studies in the Management Sciences*, 14, 209-228.

Franck, R., Melese, F. & Dillard, J. (2006). *A transactions cost economics approach to defense acquisition management.* Monterey, CA: Naval Postgraduate School.

GAO. (2005, November 15). *Acquisition outcomes—A case for change. Schinasi, Katherine V. Congressional Testimony* (GAO 06-257T DOD). Washington, DC: GAO.

GAO. (1999, July 30). *Best practices: Better management of technology development can improve weapon system outcomes* (GAO/NSIAD-99-162). Washington, DC: GAO.

GAO. (2002). *Best practices—Capturing design and manufacturing knowledge early improves acquisition outcomes* (GAO-02-701). Washington, DC: GAO.

GAO. (1997, October). *Brilliant Antiarmor Submunition opportunity exists to conduct critical test prior to production decision* (GAO/NSIAD 98-16). Washington, DC: GAO.

GAO. (1996, September). *Javelin is not ready for multiyear procurement* (GAO/NSIAD-96-199). Washington, DC: GAO.

GAO. (1998, February). *Weapons Acquisition Requires Changes in DOD's Environment* (GAO/NSIAD-98-56). Washington, DC: GAO.

GAO. (1998, March). *DEFENSE ACQUISITION: Improved Program Outcomes Are Possible* (GAO/T-NSIAD-98-123). Washington, DC: GAO.

GAO. (1999, March). *DEFENSE ACQUISITION: Best Commercial Practices Can Improve Program Outcomes* (GAO/T-NSIAD-99-116). Washington, DC: GAO.

GAO. (2000, April). *DEFENSE ACQUISITION: Employing Best Practices Can Shape Better Weapon System Decisions* (GAO/T-NSIAD-00-137). Washington, DC: GAO.

GAO. (2001, March). *BEST PRACTICES: Better Matching of Needs and Resources Will Lead to Better Weapon System Outcomes* (GAO-01-288). Washington, DC: GAO.

GAO. (2003, February). *BEST PRACTICES: Setting Requirements Differently Could Reduce Weapon Systems' Total Ownership Costs* (GAO-03-57). Washington, DC: GAO.

GAO. (2003, November). *DEFENSE ACQUISITIONS: DOD's Revised Policy Emphasizes Best Practices, but More Controls Are Needed* (GAO-04-53). Washington, DC: GAO.

Greene, H., Col. (2007, March). PM Army Battle Command. Presentation at NPS.

Hauptman, O., & Hirji, K.K. (1996). The influence of process concurrency on project outcomes in product development: An empirical study of cross-functional teams. *IEEE Transactions on Engineering Management,* 43(2), 153-178.

Hawthorne, S., & Lush, R. (2002, August). Evolutionary acquisition and spiral development. *Crosstalk.* Retrieved from http://www.stsc.hill.af.mil/crosstalk/2002/08/easd.html 14 Jan 2007.

Heylighen F. (1997). The growth of structural and functional complexity during evolution. In F. Heylighen & D. Aerts (Eds.), *The evolution of complexity*. (Kluwer, Dordrecht: Publishers).

Ibrahim, R. (2005) *Discontinuity In Organizations: Impacts Of Knowledge Flows On Organizational Performance*. Doctoral Thesis, Stanford University.

Joglekar, N. and Ford, D., (2005) Product Development Resource Allocation with Foresight, European Journal of Operational Research. Vol. 160, No. 1, pp.72-87. Jan., 2005.

Kerr, S. (Ed.) (1979). *Organizational behavior.* Columbus, OH: Grid Publishing.

Kim, Y. (2002, June). *A decomposition-based approach for the integration of product development and manufacturing system design*. Boston, MA: Massachusetts Institute of Technology.

Knox, W. (1999, September). *Presentation to the Senior Service College Fellowship Program.* Carlisle, PA.

Krishnan, V. (1996). Managing the simultaneous execution of coupled phases in concurrent product development. *IEEE Transactions on Engineering Management*, 43(2), 210-217.

Langley, M. (2006, December 22). Inside Mulally's war room: A radical overhaul for Ford. *Wall Street Journal.*

Lee, Z, Ford, D. and Joglekar, N. "Resource Allocation Policy Design for Reduced Project Duration: A Systems Modeling Approach," Systems Research and Behavioral Science. In press.

Lindblom, C. E. (1959). The science of muddling through. *Public Administration Review*, 19 (Spring); (Reprinted in Kramer, F.A. (Ed.). (1977). *Perspectives on Public Bureaucracy* (2nd ed.) (pp. 132-150). Cambridge, Massachusetts: Winthrop Publishers).

Lockheed Martin (2001, April 23). *Lockheed Martin Press Release*. Retrieved from http://www.fas.org/man/dod-101/sys/land/docs/man-la-atacms-010423.htm

Lorell, M. A., Lowell, J. F., Younossi, O. (2006). Evolutionary acquisition: Implementation challenges for defense space programs. Rand Corporation.

Lumb, M.D. (2004, February 18-19). *Meeting the security challenges of the 21st century.* Briefing, Second Annual Defense Acquisition University-South Contracting Conference and Exposition. Retrieved February 5, 2007, from http://www.dau.mil/regions/South/documents/DoD%205000%20Series%20Changes%20(Mr.%20Mark%20Lumb).pdf.

Lyneis, J. M., Cooper, K. G., & Els, S. A. (2001). Strategic management of complex projects: a case study using system dynamics. *System Dynamics Review*, 17(3), 237-260.

Lyneis, JM and Ford, DN, (2007) "System Dynamics Applied to Project Management: A Survey and Directions for Future Research". Available from authors.

Lyons, J., Long, D., & Chait, R. (2006, July). *Critical technology events in the development of the Stinger and Javelin missile systems: Project hindsight Revisited*, Washington DC: National Defense University.

Mahmoud, M. S., Al-Muthairi, N.F. (1994). Design of robust controllers for time-delay system. *IEEE Transactions on Automatic Control*, 39(5), 995-999.

Matthews, D. Col. (Ret.), former PM of the Army Tactical Missile System. (2006, December). [Interview with authors]. Naval Postgraduate School.

Mooz, H., Forsberg, K., & Cotterman, H. (2005). *Visualizing project management* (3rd ed.)

Moses, J. (2002). *Complexity and flexibility.* MIT ESD unpublished working paper. Retrieved from http://esd.mit.edu/WPS/esd-wp-2000-02a.html#view1 14 Jan 2007

Neave, H. R. (2000, March 2). *The deming dimension—Management for a better future*. Inaugural Professorial Lecture, W. Edwards Deming Professor of Management Nottingham Business School, The Nottingham Trent University.

Nissen, M.E., & Buettner, R.R. (2004). Computational experimentation with the virtual design team: Bridging the chasm between laboratory and field research in C2. 10th International Command and Control Research and Technology Symposium, Standford, CA.

Perry, W. (1995, May 10). *IPPD definition and key tenets* (Secretary of Defense Memorandum). Washington, DC: DoD.

Pich, M. T., Loch, C.H., & De Meyer, Ar. (2002, August 8). On uncertainty, ambiguity, and complexity in project management. *Management Science*, 48(8), 1008-1023.

Project Management Institute. *Project Management Body of Knowledge (PMBOK)*. (2000)

Project Management Institute. *Project Management Body of Knowledge (PMBOK)*. (2004)

RAND. (2005). *Acquisition reform: Are we there yet?* (RAND MG291) Santa Monica, CA.

Ravindran, A. Phillips, D. & Solberg, J. (1987) Operations research principles and practice (2nd ed.). New York: Wiley.

Redstone Arsenal Historical Information. (2007). Retrieved DATE from http://www.redstone.army.mil/history/netstorm/chapter3.html

Reed, M. Colonel, US Army (Ret), Contractor in support of PM Tactical Battle Command. (2006, December 16). [Interview]. “They call it soft-ware for a reason.”

Rockwell, T. (1992). *The Rickover effect: How one man made a difference.* Annapolis, MD: Naval Institute Press.

Rodrigues, A., & Williams, T. M. (1997). System dynamics in project management: assessing the impacts of client behavior on project performance. *Journal of the Operational Research Society*, 49, 2-15

Rosenthal, S.R. (1992). *Effective product design and development*. Homewood, IL: Business One Irwin.

Roy, R. S. (2006, June 5). Spiral development, real options, and other development methodologies. Presentation at the Center for Strategic and International Studies. Washington DC. Retrieved from http://www.csis.org/component/option,com_csis_events/task,view/id,985/

Shaver, R., Lynch, K.F., Amouzegar, M.A., & Snyder, D. (2005). *Unmanned aerial vehicle end-to-end support considerations* (MG-350-AF). RAND. Retrieved from http://www.rand.org/publications/MG/MG350/.

Simon, H.A. (1962).*The architecture of complexity*, Cambridge, MA: MIT Press.

Simon, H.A. (1981).*The sciences of the artificial*. (2nd ed.). Cambridge, MA: MIT Press.Smith, R. P., Eppinger, S. D., 1997b. Identifying controlling features of engineering design iteration. Management Science, 43(3) 276-293.

Smith, P. & Reinertsen, D. (1998). Developing Products in Half the Time*,* Van Nostrand Reinhold, (2nd Edition)

Sterman. J. D. (2000). Business dynamics, system thinking and modeling for a complex world. New York: Irwin McGraw-Hill.

STINET. Retrieved from http://stinet.dtic.mil/oai/oai?&verb=getRecord &metadataPrefix=html&identifier=ADA122234

Sylvester, R. K., & Ferrara, J. A. (2003). Conflict and ambiguity implementing evolutionary acquisition. *Acquisition Review Quarterly*, Winter.

Udwadia, F.E., Bremen, H.F., Kumar, R., Hosseini, M. (2003). Time delayed control of structural systems. *Earthquake Engineering and Structural Dynamics*, 32, 495-535.

US Air Force. Retrieved February 25, 2007, from http://www.af.mil/factsheets/.

US Army RDT&E. (2004, February). Budget item justification R-2 exhibit. Retrieved from http://www.dtic.mil/descriptivesum/Y2005/Army/0604768A.pdf

United States Code. (2002). Title 10, Subtitle A, Part IV, Chapter 144, § 2430.

United States Code. (Now in statute: amended in 2006). Title 10, Chapter 139 Sec. 2366a.

USD(AT&L). (2000, October 23). *Department of Defense Instruction 5000.2*. Operation of the Defense Acquisition System, October 23, 2000.

USD(AT&L). (2003a, May 12). *Department of Defense Directive 5000.1*. The Defense Acquisition System. Washington, DC: author.

USD(AT&L). (2003b, May 12). *Department of Defense instruction 5000.2.* Operation of Defense Acquisition System. Washington, DC: author.

Washington Technology. (June 9, 2003). Retrieved from http://www.washingtontechnology.com/news/18_5/cover-stories/20872-1.html 14 Jan 2007.

Watson, R., & Pollack, J. (2005). Modular interdependency in complex dynamical systems. MIT. *Artificial Life* 11, 445-457.

Wikipedia. (2007). Black Hawk. Retrieved February 25, 2007, from http://en.wikipedia.org/wiki/UH-60_Black_Hawk.

Woodward, J. (1958). *Management and technology.* London: HMSO.

Wysocki, R.K. (2003). *Effective project management* (3rd ed.). Indianapolis, IN: Wiley.

Zolin, R. & Dillard, J.T. (2005, May). From market to clan: How organizational control affects trust in defense acquisition. *Proceedings.* Second Acquisition Research Symposium. Monterey, CA: Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix 1. UH-60 Series Helicopter Variants Introduced Between 1979-2007

- **UH-60A Black Hawk** - Original U.S. Army version deployed in **1979**, carrying a crew of four and up to 11 passengers. Equipped with T-700-GE-700 engines.
- **UH-60A RASCAL** - NASA-modified version for the Rotorcraft-Aircrew Systems Concepts Airborne Laboratory.
- **EH-60A Black Hawk** - Modified electrical system and stations for two electronic systems mission operators.
- **MH-60A Black Hawk** - Modified with additional avionics, precision navigation system, FLIR and air-to-air refueling capability. Equipped with T-700-GE-701 engines.
- **YEH-60B Black Hawk** - UH-60A modified for special radar and avionics installations, prototype for stand-off target acquisition system.
- **SH-60B Seahawk** - The United States Navy's sea-going version. Based on UH-60A but with Mark III avionics. Equipped with T-700-GE-401 engines.
- **UH-60C Black Hawk** - Modified version for C2 missions.
- **EH-60C Black Hawk** - UH-60A modified with special electronics equipment and external antenna.
- **VH-60D Nighthawk** - VIP-configured HH-60D, used for Presidential transport. T-700-GE-401 engines.
- **SH-60F Seahawk** - Navy upgrade version, received in 1988, equipped with dipping sonar.
- **NSH-60F Seahawk** - Modified SH-60F to support the VH-60N Cockpit Upgrade Program.
- **HH-60G Pave Hawk** - Modified UH-60A primarily designed for combat search and rescue. It is equipped with a rescue hoist with a 200 ft (60.96 m) cable that has a 600 lb (270 kg) lift capability, and a retractable in-flight refueling probe.
- **MH-60G Pave Hawk** - Special Operations version, equipped with long-range fuel tanks, air-to-air refueling capability, FLIR, improved radar. T-700-GE-700/701 engines.
- **HH-60H Sea Hawk** - Modified SH-60F with both offensive and defensive weaponry. T-700-GE-401 engines.

- **HH-60J Jayhawk** - The United States Coast Guard version, equipped with a rescue hoist with a 200 ft (60.96 m) cable that has a 600 lb (270 kg) lift capability.
- **MH-60K Blackhawk** - Special operations modification,
- **UH-60L Black Hawk** - UH-60A with upgraded T-700-GE-701C engines, improved durability gearbox, and additional vibration absorbers.
- **EUH-60L** - Modified with additional mission electronic equipment for Army Airborne C2.
- **EH-60L Black Hawk** - EH-60A with major mission equipment upgrade.
- **HH-60L** - UH-60L extensively modified in **1989** with medical mission equipment. Components include an external rescue hoist, integrated patient configuration system, and aircrew positions relocated to the back of the cabin.
- **MH-60L Direct Action Penetrator (DAP)** - Special operations modification, operated by the 160th Special Operations Aviation Regiment. It is capable of being armed with 30mm chain gun and 2.75" rockets, as well as M134D gatling guns operated as door guns or fixed forward.
- **UH-60M Black Hawk** - UH-60L upgraded with improved design "wide chord" rotor system, T-700-GE-701D Engines, improved durability gearbox, integrated Vehicle Management Systems (IVHMS) computer, and modern "Glass Cockpit" flight instrument suite. Planned to replace all UH-60A and L aircraft with the U.S. Army.
- **HH-60M** - UH-60A with medical mission equipment.
- **VH-60N Nighthawk** - Modified HH-60D used for Presidential transport.
- **UH-60Q Black Hawk** - UH-60A modified for medical evacuation.
- **YMH-60R Sea Hawk** - Prototype for MH-60R. T-700-GE-701C engines.
- **MH-60R Sea Hawk** - Modified SH-60B for multiple mission use. T-700-GE-401 engines.
- **SH-60R Sea Hawk** - Modified SH-60B with improved radar and sonar systems.
- **NSH-60R Sea Hawk** - U.S. Navy special testing version. T-700-GE-701C engines.
- **CH-60S Sea Hawk** - Upgrade of UH-60L and SH-60R for cargo transport.
- **MH-60S** - Navy medical evacuation and ship replenishment mission equipped. T-700-GE-401 engines.

(DoD, 2004, May 12; Wikipedia, 2007)

Appendix 2. C-130 Hercules Aircraft Variants Introduced Between 1956-2007

- The **C-130A** entered initial production with four Allison T56-A-11 or -9 turboprops engines. A total of 219 were ordered and deliveries began in **December 1956**.
- The **C-130B** introduced Allison T56-A-7 turboprops and the first of 134 entered Air Force service in **May 1959**.
- The **C-130E** was introduced in **August of 1962** with a production run of 389, using the same Allison T56-A-7 engine, but adding two 1,290 gallon external fuel tanks and an increased maximum takeoff weight capability.
    - **Speed:** 345 mph at 20,000 feet
    - **Ceiling:** 19,000 feet with 42,000 pounds payload
    - **Maximum Allowable Payload**: 42,000 pounds
    - **Range at Maximum Normal Payload:** 1,150 miles
- The **C-130H** was introduced in **June 1974** as the first of 308 with the more powerful Allison T56-A-15 turboprop engine delivering 4,591prop shaft horsepower. Nearly identical to the C-130E externally, the new engine brought major performance improvements to the aircraft.
    - **Speed:** 366 mph at 20,000 feet
    - **Ceiling:** 23,000 feet with 42,000 pounds payload.
    - **Maximum Allowable Payload**: 42,000 pounds
    - **Range at Maximum Normal Payload:** 1,208 miles
- The **C-130J** entered the inventory in **February 1999**. With a six-bladed composite propeller coupled to a 4,700 horsepower Rolls-Royce AE2100D3 turboprop engine, the C-130J brings substantial performance improvements over all previous models.
    - **Speed:** 417 mph at 22,000 feet
    - **Ceiling:** 28,000 with 42,000 pounds payload
    - **Maximum Allowable Payload**: 42,000 pounds
    - **Range at Maximum Normal Payload:** 2,071 miles
- The **C-130J-30**, a stretch version with a 15-foot fuselage extension. To date, the Air Force has taken delivery of 37 C-130J aircraft from Lockheed Martin Aeronautics Company.
    - **Speed:** 410 mph at 22,000 feet

- **Ceiling:** 26,000 feet with 44,000 pounds payload.
- **Maximum Allowable Payload**: 44,000 pounds
- **Range at Maximum Normal Payload:** 1,956 miles

- The **AC-130H/U Gunship** is a heavily armed, incorporating side-firing cannons integrated with sophisticated sensor, navigation and fire control systems to provide surgical firepower or area saturation during extended loiter periods, at night and in adverse weather. The AC-130U (**deployed 1995**) employs synthetic apertures strike radar for long-range target detection and identification. The navigational devices include the inertial navigation systems and global positioning system. The AC-130U employs the latest technologies and can attack two targets simultaneously. It also has twice the munitions capacity of the AC-130H (**deployed 1972**).
- The **MC-130E** Combat Talon I and MC-130H Combat Talon II provide infiltration, exfiltration and resupply of special operations forces and equipment in hostile or denied territory.
- The **MC-130P** Combat Shadow features improved navigation, communications, threat detection and countermeasures systems. The Combat Shadow fleet has a fully integrated inertial navigation and global positioning system, and night vision goggle compatible interior and exterior lighting.
- The **MC-130W** (**deployed 2006**) is a highly modified C-130H featuring improved navigation, threat detection and countermeasures, and communication suites, with air refuel capability for special operations helicopters.
- The **WC-130H** Hercules is configured with computerized weather instrumentation for penetration of severe storms to obtain data on storm movements, dimensions and intensity. The WC-130B became operational in **1959**, the E model in **1962**, followed by the H model in **1964**. Only the H model is currently in operation. The WC-130J, currently in testing, is scheduled to replace the WC-130H.

(US Air Force, 2007, February 25)

Not an inclusive list; the authors have found a total of 24 Hercules C-130 variants across the US Air Force and Navy (DoD, 2004, May 12).

Initial Distribution List

1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944; Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library, Code 013 Naval Postgraduate School, Monterey, CA 93943-5100	2
3.	Research Office, Code 09 Naval Postgraduate School, Monterey, CA 93943-5138	1
4.	Robert N. Beck Dean, GSBPP 555 Dyer Road, Naval Postgraduate School, Monterey, CA 93943-5000	1
5.	Keith F. Snider Associate Professor, GB/Sk 555 Dyer Road, Naval Postgraduate School, Monterey, CA 93943-5000	1
6.	James B. Greene Acquisition Chair, GB/Jg 555 Dyer Road, Naval Postgraduate School, Monterey, CA 93943-5000	1
7.	Bill Gates Associate Dean for Research, GB/Gt 555 Dyer Road, Naval Postgraduate School, Monterey, CA 93943-5000	1
8.	John Dillard Senior Lecturer, GB/Dj 555 Dyer Road, Naval Postgraduate School, Monterey, CA 93943-5000	1
9.	David N. Ford Zachry Department of Civil Engineering Texas A&M University, College Station, TX 77843-3136	1
10.	Karey L. Shaffer Program Manager, Acquisition Research Program, GB/Ks 555 Dyer Road, Naval Postgraduate School, Monterey, CA 93943-5000	1

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.nps.navy.mil/gsbpp/acqn/publications

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2006 Sponsored Acquisition Research Topics

Acquisition Management

- Software Requirements for OA
- Managing Services Supply Chain
- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Portfolio Optimization via KVA + RO
- MOSA Contracting Implications
- Strategy for Defense Acquisition Research
- Spiral Development
- BCA: Contractor vs. Organic Growth

Contract Management

- USAF IT Commodity Council
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Navy Contract Writing Guide
- Commodity Sourcing Strategies
- Past Performance in Source Selection
- USMC Contingency Contracting
- Transforming DoD Contract Closeout
- Model for Optimizing Contingency Contracting Planning and Execution

Financial Management

- PPPs and Government Financing
- Energy Saving Contracts/DoD Mobile Assets
- Capital Budgeting for DoD
- Financing DoD Budget via PPPs
- ROI of Information Warfare Systems

- Acquisitions via leasing: MPS case
- Special Termination Liability in MDAPs

Logistics Management

- R-TOC Aegis Microwave Power Tubes
- Privatization-NOSL/NAWCI
- Army LOG MOD
- PBL (4)
- Contractors Supporting Military Operations
- RFID (4)
- Strategic Sourcing
- ASDS Product Support Analysis
- Analysis of LAV Depot Maintenance
- Diffusion/Variability on Vendor Performance Evaluation
- Optimizing CIWS Lifecycle Support (LCS)

Program Management

- Building Collaborative Capacity
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Terminating Your Own Program
- Collaborative IT Tools Leveraging Competence

A complete listing and electronic copies of published research within the Acquisition Research Program are available on our website: www.acquisitionresearch.org

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org